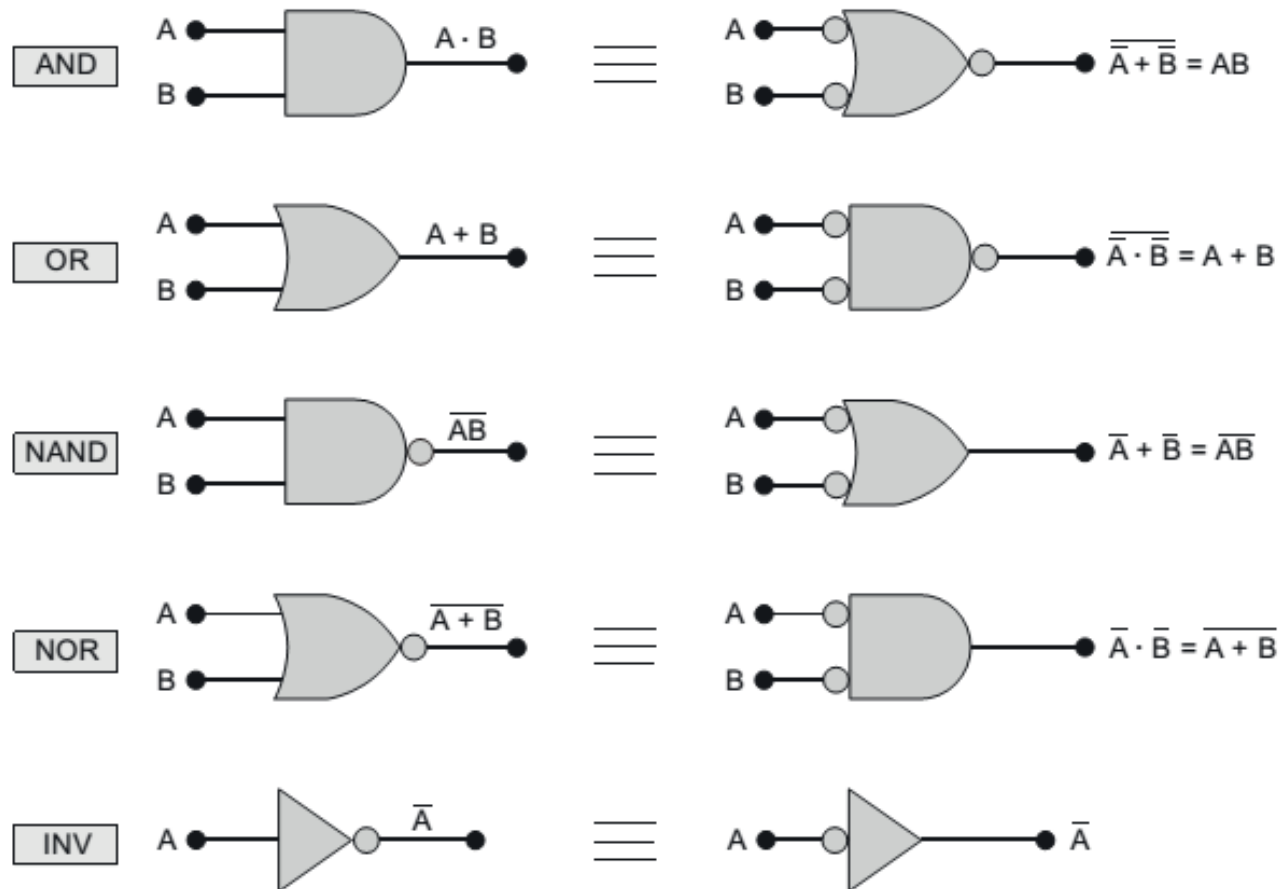


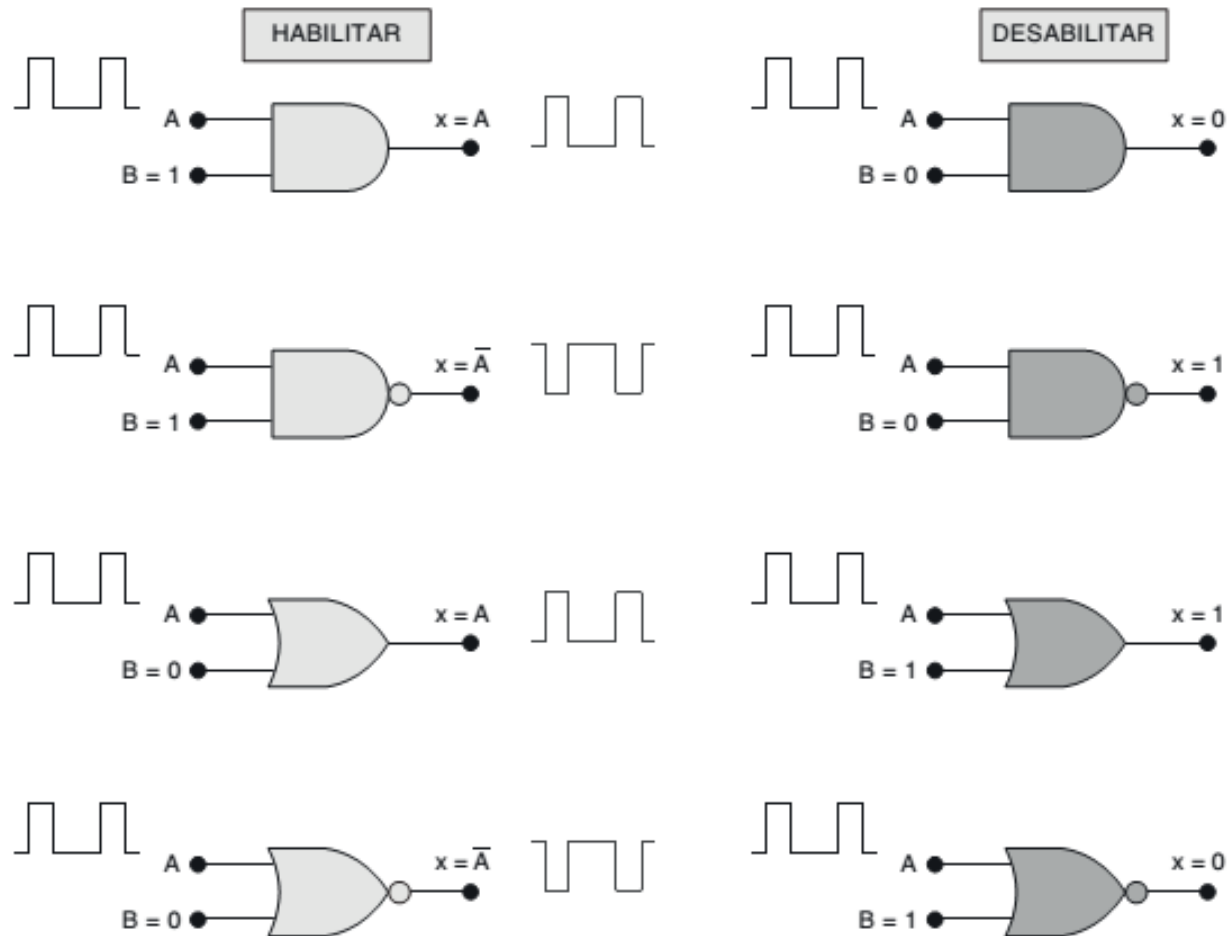
Simbologia Alternativa para Circuitos Lógicos



Circuitos Para Habilitar e Desabilitar

- As portas lógicas podem ser utilizadas para controlar a passagem de um sinal lógico de entrada para a saída.
- Numa porta lógica de 2 entradas (A,B), uma delas atua como o sinal que quer ser transmitido e a outra entrada como controle para esta saída.
- O nível lógico da entrada de controle determina se o sinal de entrada está habilitado ou desabilitado para atingir a saída.

Circuitos Para Habilitar e Desabilitar



Circuitos Digitais

- Os circuitos digitais são divididos em duas famílias importantes:
 - Circuitos combinacionais: as saídas dependem exclusivamente das variáveis de entrada. Esses circuitos **não possuem memória**, ou seja, não armazenam nem dependem de entradas ou saídas passadas. Ex. cadeado de mala.
 - A saída z do circuito no instante t depende somente de suas entradas $x_1, x_2, x_3 \dots x_n$ no instante t :
$$z(t) = f(x_1(t), x_2(t), x_3(t) \dots x_n(t))$$
 - Circuitos sequenciais: as saídas dependem das variáveis de entrada, bem como de seus estados anteriores, que **permanecem armazenados**. Ex. cadeado de cofre.
 - De forma geral, os circuitos sequenciais operam sob o comando de uma sequência de pulsos denominada *clock*.

Elementos de memória

- Para armazenar os valores passados das entradas e saídas, utiliza-se a noção de “estado”
 - ESTADO é representado por um conjunto de variáveis de estado que contém toda a informação relevante sobre o passado do circuito, e que permite descrever seu comportamento futuro.
 - Pergunta: como podemos armazenar uma informação quando ela não está mais presente?

Exemplo de “estado”

- Um circuito possui duas “botoeiras”, A e B, que determinam o estado da lâmpada L: acesa ($L = 1$) ou apagada ($L=0$). A lâmpada deve se acender quando A é acionada e apagar quando B for acionada. É impossível acionar as 2 botoeiras juntas. Monte a tabela verdade.

Elementos de memória

- Os circuitos básicos que implementam a função de memória são denominados biestáveis, fliflops ou latches.
 - Latches são sensíveis ao nível dos sinais de entrada
 - Flipflops são sensíveis à borda dos sinais da entrada de controle

TIPOS DE F.F. e LATCHES:

- SR
- JK
- D
- T

Circuitos síncronos e assíncronos

- Em circuitos assíncronos, as alterações nas saídas ocorrem em qualquer instante, de acordo com alterações dos valores nas entradas.
- Em circuitos síncronos, as alterações nas saídas ocorrem em instantes específicos, sincronizados com a ocorrência de um sinal numa entrada especial denominada relógio (*clock*).

Relógio (Clock)

- **Nível ativo: Clock ativo em Alto ou Baixo:**
 - Se Ativo em Alto, as transições de estado ocorrem quando o sinal de *clock* está em 1
 - Se Ativo em Baixo, as transições de estado ocorrem quando o sinal de *clock* está em 0
- **Período**
 - Intervalo de tempo entre duas transições do *clock* no mesmo sentido.
- **Frequência**
 - Inverso do Período
- *Duty-Cycle*
 - Relação entre o “tempo ativo” e o período

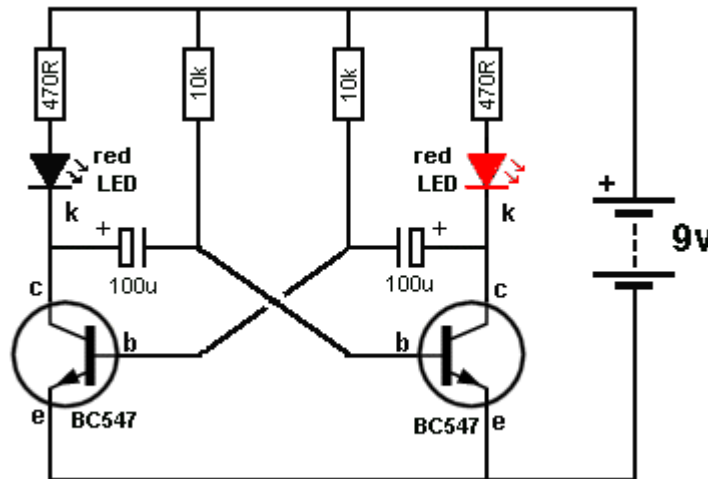
Relógio (Clock) Ativo em Nível Alto



- Período = T_{CLK}
- Frequência = $1 / T_{CLK}$
- *Duty cycle* = t_H / T_{CLK}

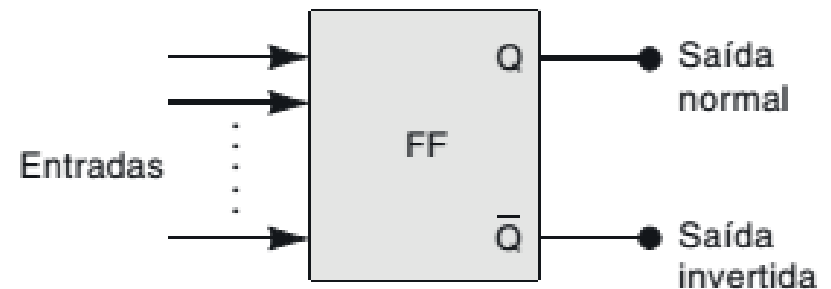
Flip-Flops

- O elemento de memória mais importante é o **flip-flop**, composto por um conjunto de portas lógicas.
- Um elemento de memória pode ser criado aplicando-se o conceito de **realimentação**.
- São utilizados para armazenar/transferir dados como: buffer de dados, interfaces, contadores, conversores serial-paralelo etc



Flip-Flops

- Entradas de controle
 - Depende do tipo de flip-flop em questão
- Saídas Q e Q'
 - Q é a saída normal do FF e Q' a saída invertida
 - Q representa o estado do FF
- Tipo SR
- Tipo JK
- Tipo D
- Tipo T



Estados de saída

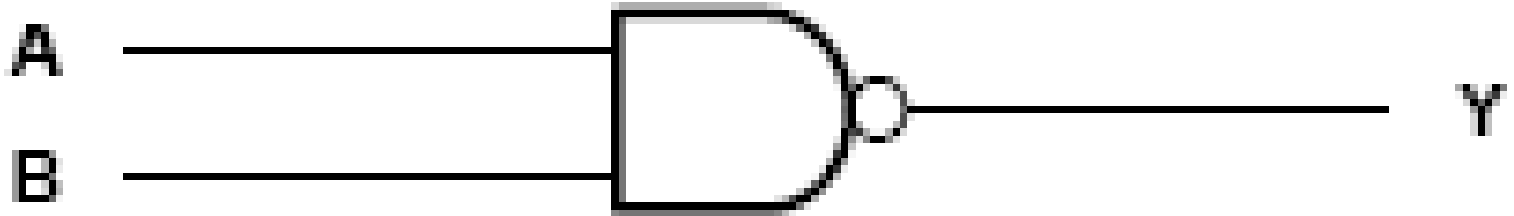
$Q = 1, \bar{Q} = 0$: chamado estado ALTO ou 1;
também chamado estado SET

$Q = 0, \bar{Q} = 1$: chamado estado BAIXO ou 0;
também chamado estado CLEAR ou RESET

Flip-Flop SR

- SET/RESET(CLEAR)
 - O circuito de um FF mais simples pode ser construído a partir de duas portas NAND ou duas portas NOR.
 - $Q = 1$ “*setar*” o flip-flop
 - $Q = 0$ “*resetar*” o flip-flop
 - Inicialmente as entradas estão em repouso: sempre que se deseja alterar as saídas uma delas é pulsada.

NAND

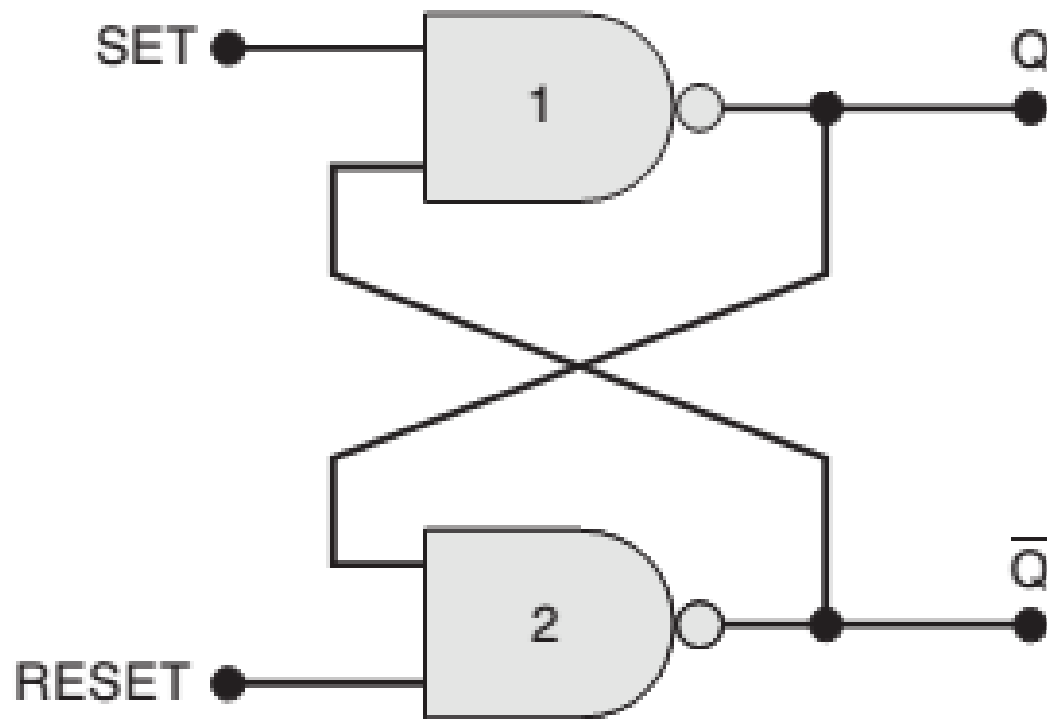


A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

Latch com NAND

- As duas portas NAND são interligadas de modo cruzado, de modo que a saída da NAND1 seja conectada a uma das entradas da NAND2 e vice-versa.
- A configuração de circuito dá a realimentação necessária para produzir a função de memória.
- As saídas Q e \bar{Q} são as saídas do Latch.
- A entrada SET seta a saída Q para o nível 1 e a entrada Reset seta o Q para o nível 0.

Latch com NAND

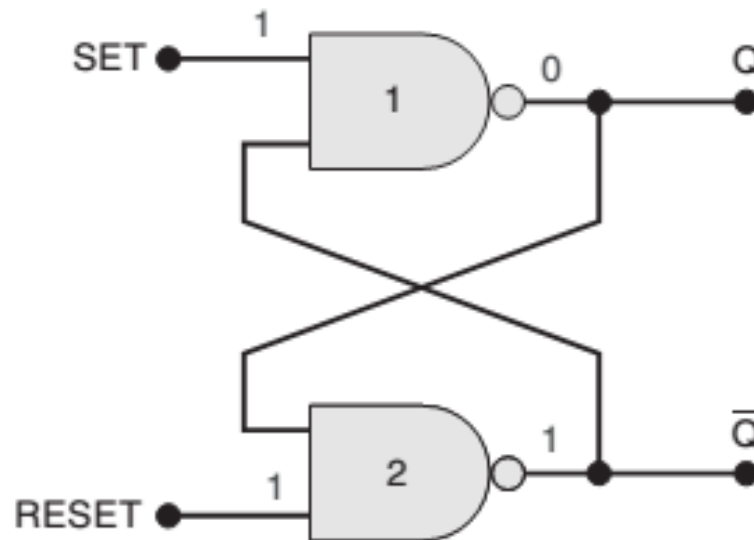


Latch com NAND

- Entradas em repouso (nível ALTO)
- Sempre que se deseja alterar as saídas, uma das entradas é pulsada (nível BAIXO).
- Existem dois estados de saída igualmente prováveis quando $SET=RESET=1$.
- Quando energizado não é possível prever o estado inicial da saída do FF se as entradas $SET=RESET=1$
- Existem chances iguais de o estado inicial da saída ser $Q=0$ ou $Q=1$
 - Dependência de fatores como atrasos internos de propagação, capacitâncias parasitas e carga externa.

Latch com NAND

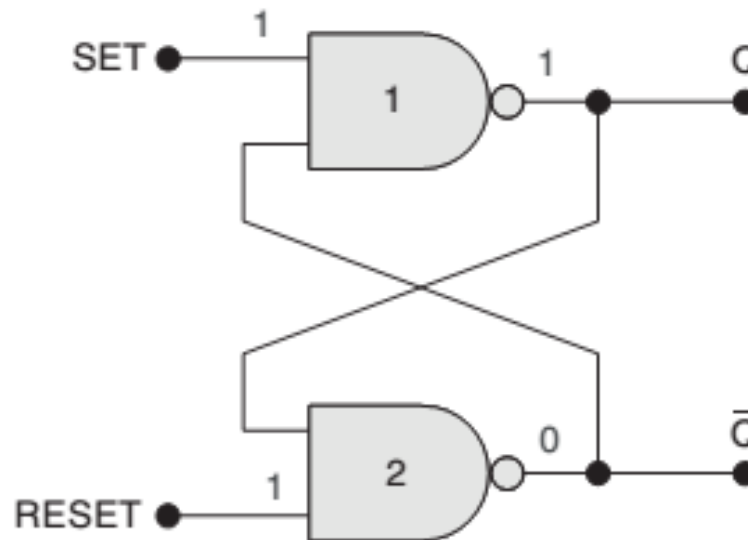
- Se $SET=Reset=1$ e $Q_o=0$ então NAND2 produz a saída $\bar{Q}=1$ e consequentemente NAND1 dá saída $Q=0$



A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

Latch com NAND

- Se $SET=Reset=1$ e $Q_0=1$ então NAND2 produz a saída $\bar{Q}=0$ e consequentemente NAND1 dá saída $Q=1$



A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

Latch com NAND

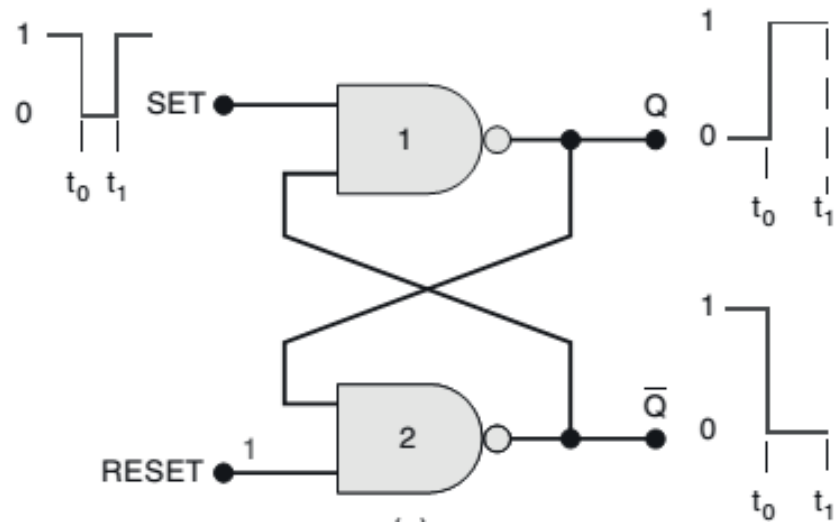
- Se um *latch* tiver que iniciar em um estado particular para garantir a operação adequada de um circuito, ele não deve ser iniciado com $SET=RESET=1$, ou seja, terá de ser colocado no estado desejado (Q).
- Aplicar pulso apropriado na entrada SET ou RESET no início da operação do circuito.

“Setando” o *Latch*

- Análise quando $Q_0 = 0$ ao energizar
 - Quando $SET=0$ no instante t_0 , saída altera para $Q=1$
 - Quando retornamos $SET=1$ no instante t_1 , valor da saída permanece $Q=1$

NAND

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

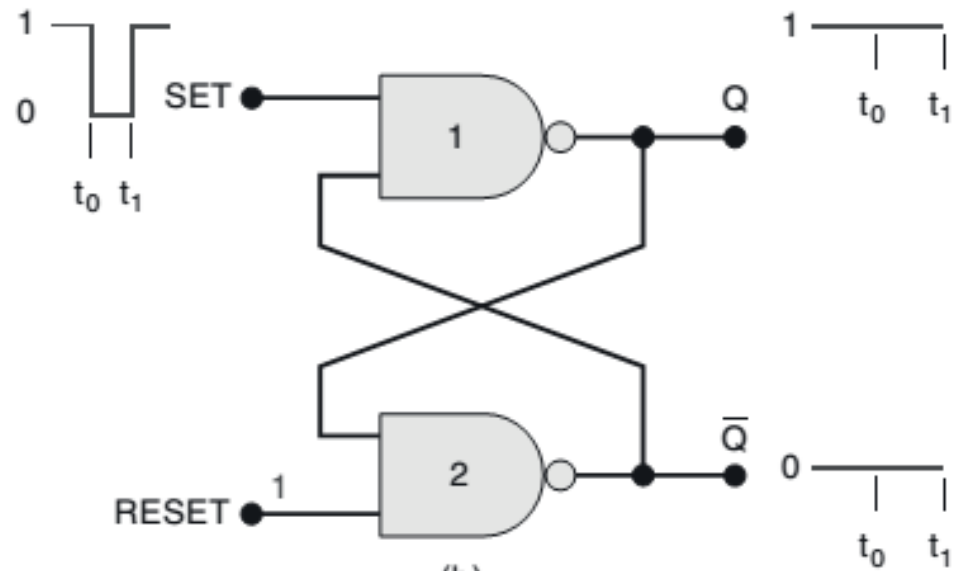


Setando o Latch

- Análise quando $Q_0 = 1$ ao energizar
 - Quando $SET=0$ no instante t_0 saída permanece $Q=1$
 - Quando retornamos $SET=1$ no instante t_1 , valor da saída permanece $Q=1$

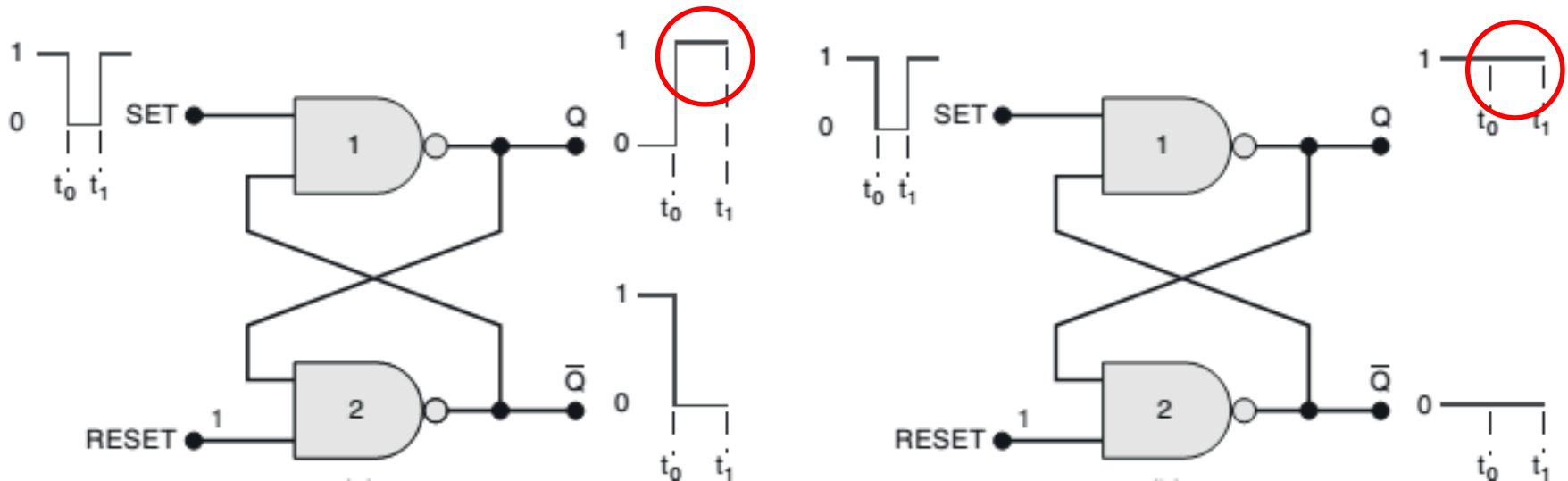
NAND

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0



Setando o Latch

- Nos dois casos anteriores a saída assume valor **Q=1** quando entrada SET é pulsada para um nível baixo.

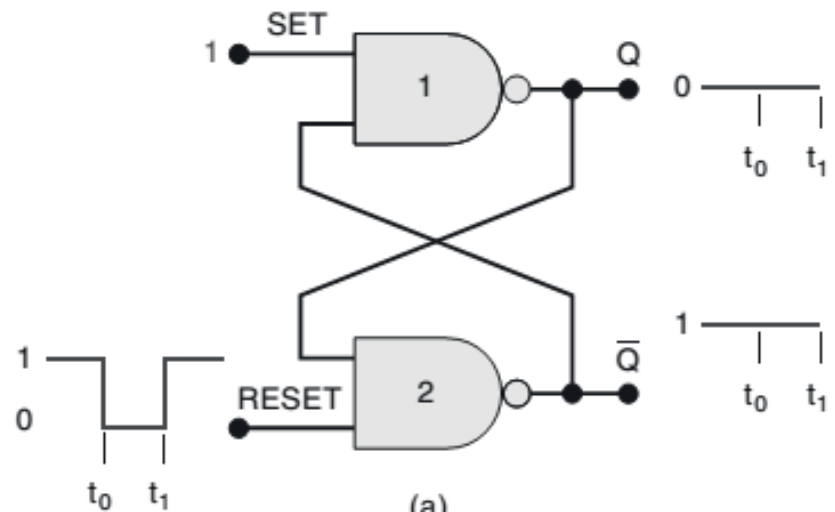


“Resetando” *Latch*

- Análise quando $Q_0 = 0$ ao energizar
 - Quando $RESET = 0$ no instante t_0 , valor da saída permanece $Q = 0$
 - Quando retornamos $RESET = 1$ no instante t_1 , valor da saída permanece $Q = 0$

NAND

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

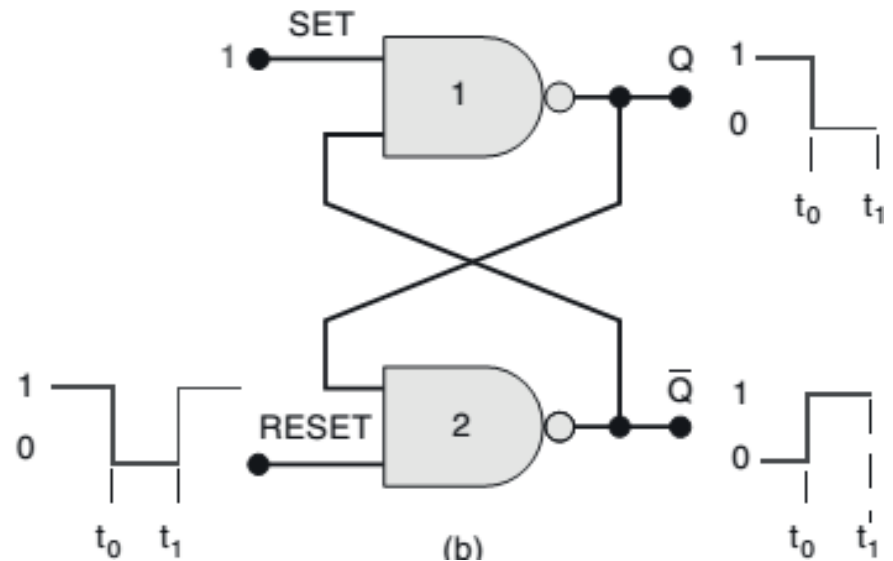


“Resetando” *Latch*

- Análise quando $Q_0 = 1$ ao energizar
 - Quando $RESET=0$ no instante t_0 , valor da saída altera para $Q=0$
 - Quando retornamos $RESET=1$ no instante t_1 , valor da saída permanece $Q=0$

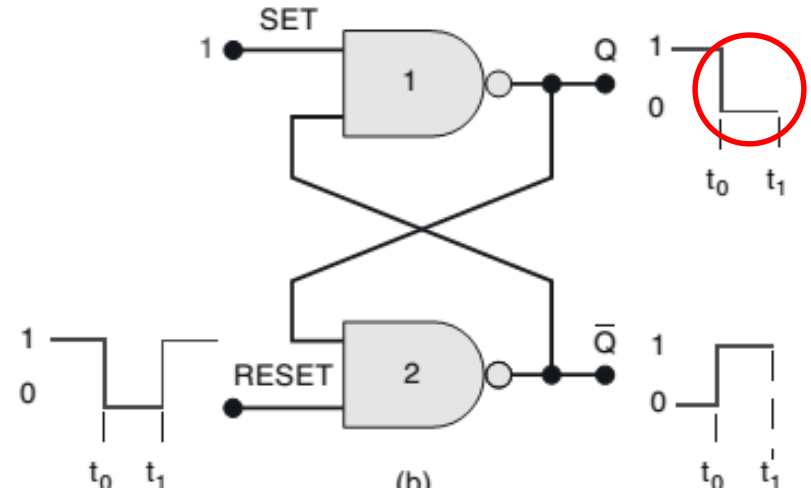
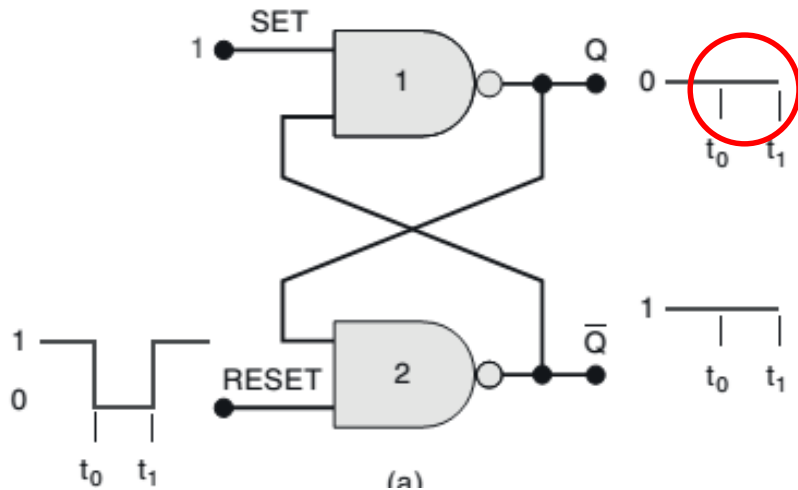
NAND

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0



“Resetando” *Latch*

- Nos dois casos anteriores a saída assume valor $Q=0$ quando entrada RESET é pulsada para um nível baixo.



Latch com NAND

Resumo

- SET=RESET=1
 - Estado normal de repouso
 - Não tem nenhum efeito na saída
 - Saída Q permanece na mesma da condição anterior
- SET=0; RESET=1 (Setar o latch)
 - Saída Q=1
 - Saída permanece Q=1 mesmo se SET=1
- SET=1; RESET=0
 - Saída Q=0
 - Saída permanece Q=0 mesmo se RESET=1

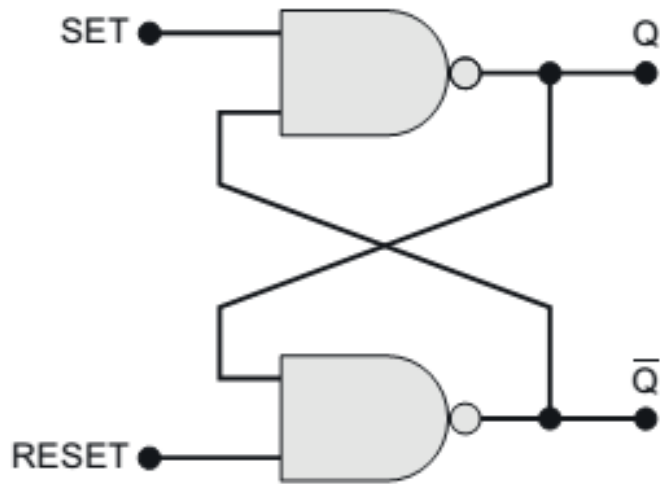
Latch com NAND

Resumo

- SET=RESET=0
 - Tenta a mesmo tempo *setar* e *resetar* o latch
 - Produz $Q=\bar{Q}=1$
 - Se as entradas retornarem ao 1 simultaneamente o resultado é imprevisível
 - **Condição inválida**

Latch com NAND

Resumo

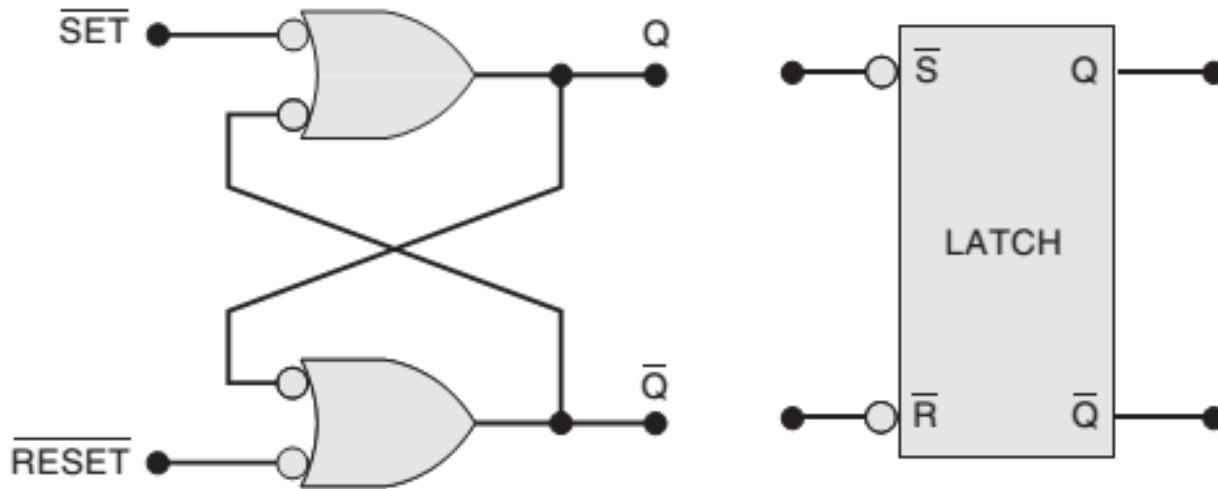


SET	RESET	Saída
0	0	Inválida*
0	1	Q=1
1	0	Q=0
1	1	Não muda

* Produz $Q=\bar{Q}=1$

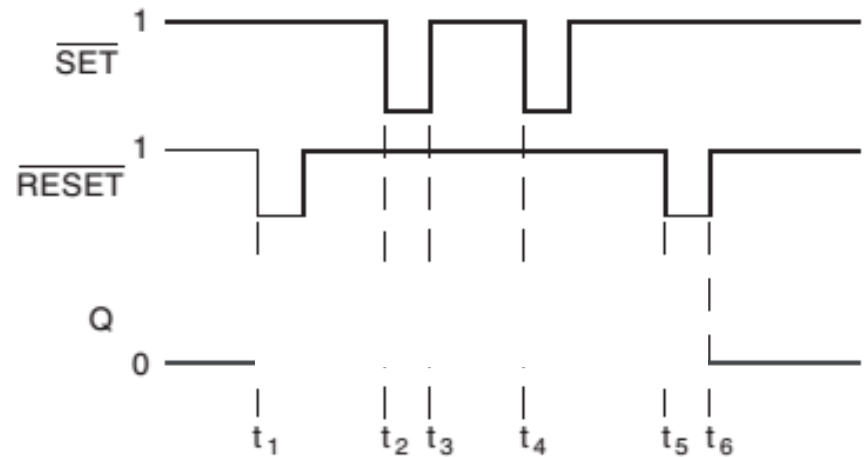
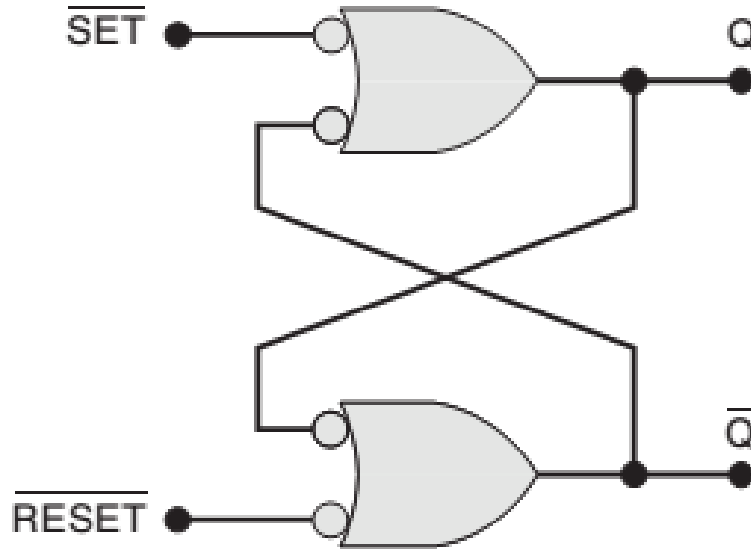
Representações Alternativas

- No *latch* com **portas NAND**, fica claro que as entradas SET e RESET **são ativas** em nível BAIXO.



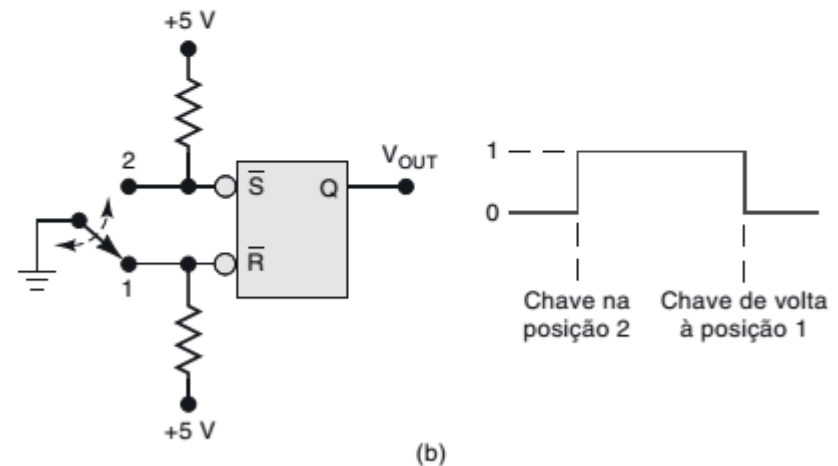
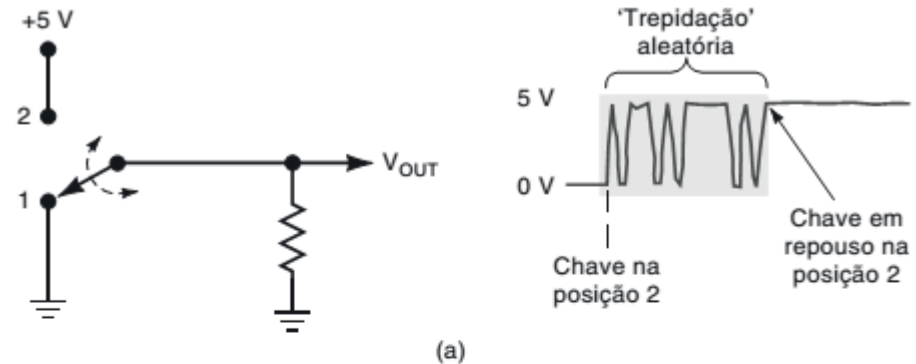
Os pequenos círculos nas entradas, assim como os nomes dos sinais \overline{SET} e \overline{RESET} , indicam o estado de ativação em nível BAIXO dessas entradas.

- Exemplo: As formas de onda na são aplicadas nas entradas do latch mostrado. Considerando que inicialmente $Q = 0$, determine a forma de onda na saída Q .



Exemplo

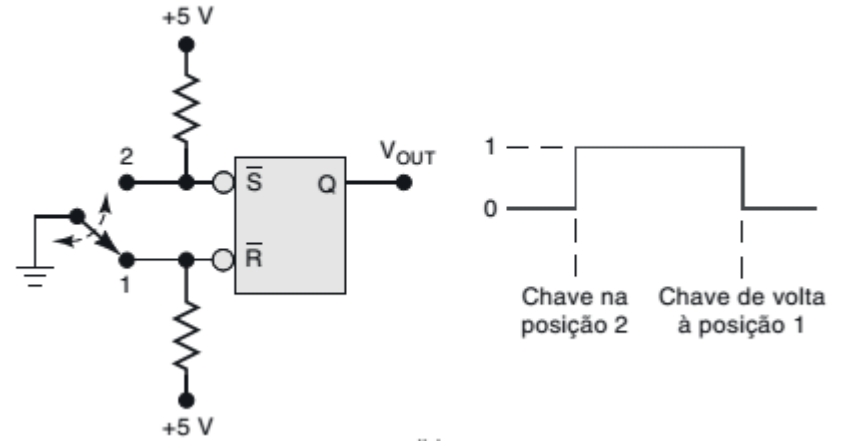
- É praticamente impossível obter uma transição 'limpa' de tensão a partir do movimento da chave da posição 1 para a 2, devido ao fenômeno da **trepidação do contato** (*contact bounce*).
- Um latch com portas NAND pode ser usado para evitar que a presença da trepidação do contato afete o sinal de saída.



(a) A trepidação de um contato mecânico gera múltiplas transições na tensão;
(b) latch NAND usado para eliminar as múltiplas transições na tensão.

Exemplo

- Descreva como podemos evitar, utilizando circuitos digitais, a **trepidação do contato** em cada uma opções de posição da chave.



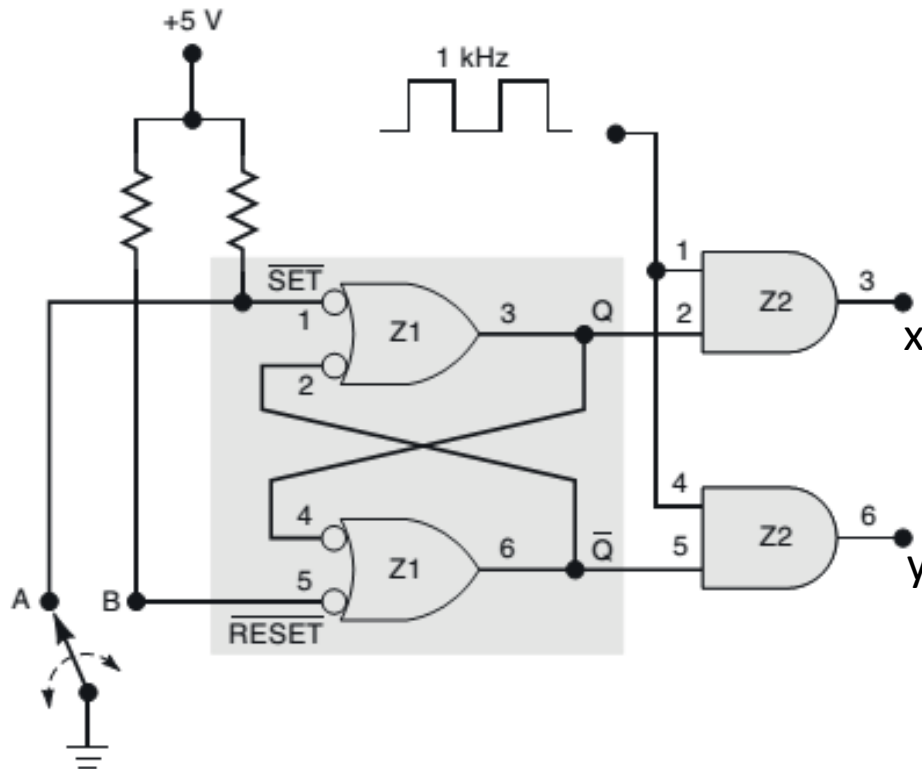
SET	RESET	Saída
0	0	Inválida*
0	1	$Q=1$
1	0	$Q=0$
1	1	Não muda

Flip-Flops

- Quando um flip-flop é ligado à alimentação, não é possível prever o estado inicial da saída...
 - ...se o SET e o RESET estão no estado “desativado”
- Para iniciar um *flip-flop* em um estado particular, é necessário colocar o estado momentaneamente ativando o SET ou o RESET no início da operação
 - Aplicação de um pulso na entrada correspondente

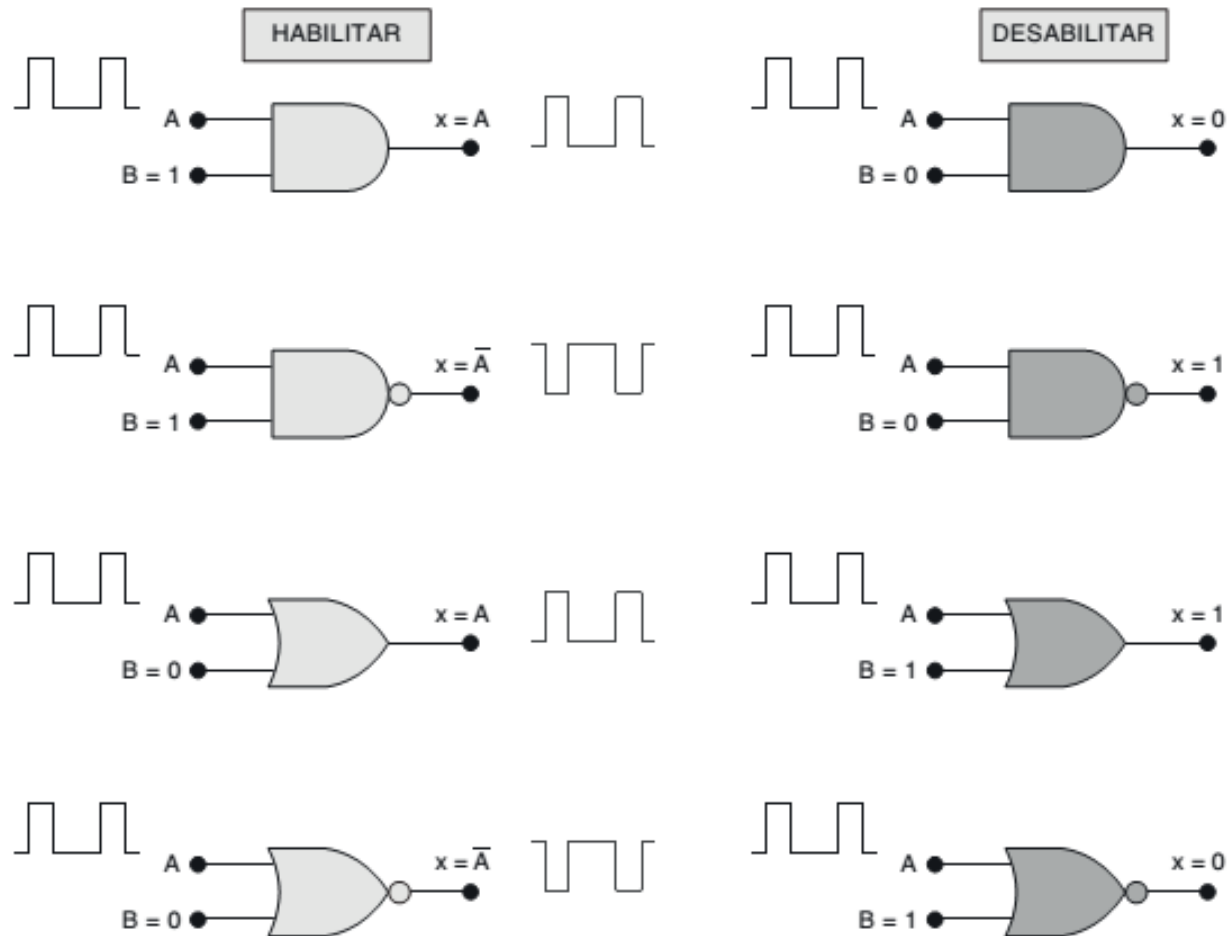
Exemplo de Aplicação

- Analise e descreva o funcionamento do circuito mostrado na Figura. Considere inicialmente que $Q=0$.



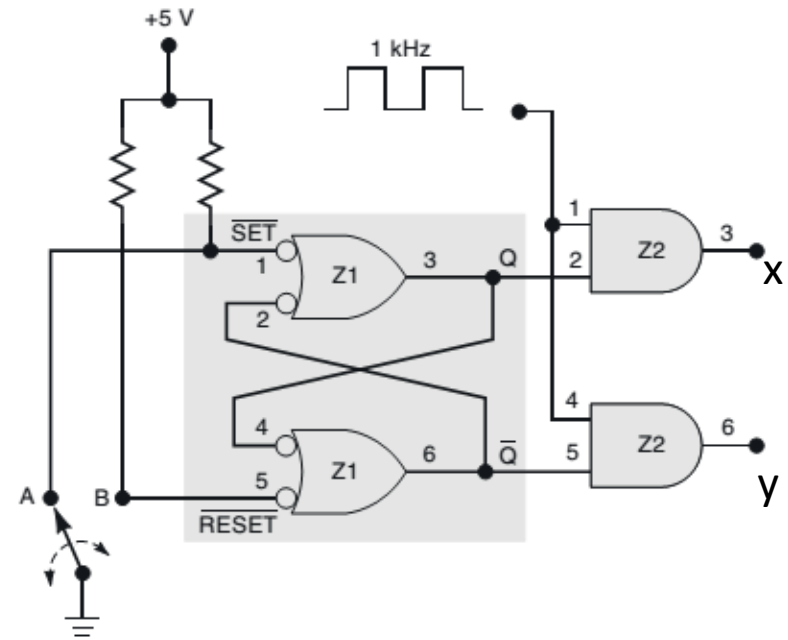
SET	RESET	Saída
0	0	Inválida*
0	1	$Q=1$
1	0	$Q=0$
1	1	Não muda

Circuitos Para Habilitar e Desabilitar

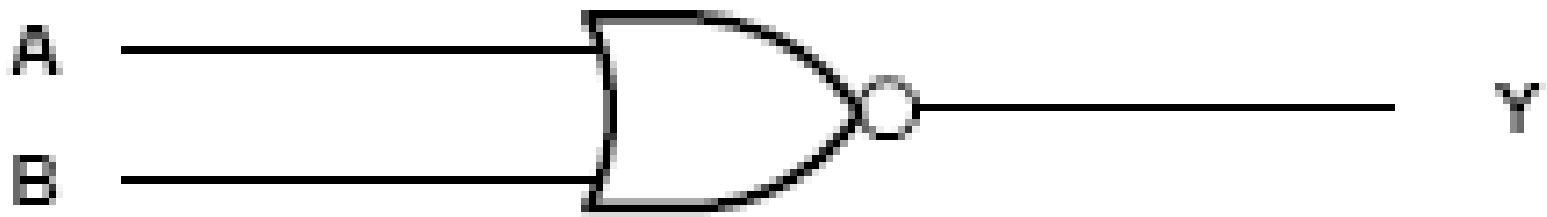


Exemplo de Aplicação

- A chave é usada para “setar” ou “resetar” o *latch* NAND, Inicialmente $Q_0=0$.
- As saídas do latch controlam a passagem de um sinal formado por pulsos retangulares com frequência de 1 kHz por meio das saídas X e Y das portas AND.
- Quando a chave é colocada na posição A, o *latch* é setado ($Q = 1$). Isso habilita os pulsos de 1 kHz a chegarem à saída X, enquanto o nível BAIXO em \bar{Q} mantém $Y = 0$.
- Quando a chave é colocada na posição B, o *latch* é resetado ($Q = 0$), mantendo $X = 0$, enquanto o nível ALTO em \bar{Q} habilita a passagem dos pulsos para Y.



NOR



A	B	S
0	0	1
0	1	0
1	0	0
1	1	0

Latch com NOR

- As entradas em repouso quando estão em nível BAIXO.
- Sempre que se deseja alterar as saídas, uma delas é pulsada (nível ALTO).
- Quando energizado não é possível prever o estado inicial da saída do FF se as entradas $SET=RESET=1$
 - Existem chances iguais de o estado inicial da saída ser $Q=0$ ou $Q=1$
 - Dependência de fatores como atrasos internos de propagação, capacitâncias parasitas e carga externa.

Latch com NOR

Inicialização do *Latch*

- Para garantir a operação adequada de um circuito, ele não deve ser iniciado com $SET=RESET=1$, ele terá de ser colocado no estado desejado ($Q=0$ ou $Q=1$).
- Aplicar pulso apropriado na entrada SET ou RESET no início da operação do circuito.

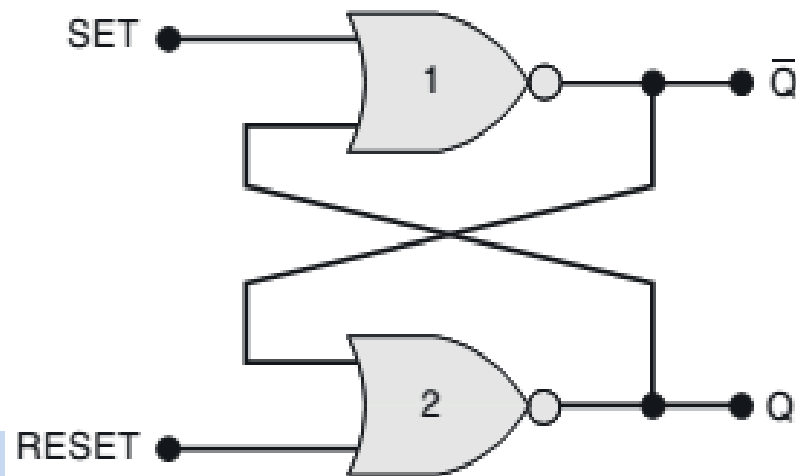
Latch com NOR

Set=0 e Reset =0

- Com $Q_0=0$ então NOR1 dá saída $\bar{Q} = 1$ e consequentemente NOR1 dá saída $Q=0$.
- Se $Q_0=1$ então NOR2 dá saída $\bar{Q} = 0$ e consequentemente NOR1 dá saída $Q=1$

NOR

A	B	S
0	0	1
0	1	0
1	0	0
1	1	0



Latch com NOR

Resumo

- $SET=RESET=0$
 - Estado normal de repouso
 - Não tem nenhum efeito na saída
 - Saída Q permanece a mesma da condição anterior
- $SET=1; RESET=0$ (Setar o latch)
 - Saída $Q=1$
 - Saída permanece $Q=1$ mesmo se $SET=0$
- $SET=0; RESET=1$
 - Saída $Q=0$
 - Saída permanece $Q=0$ mesmo se $RESET=1$

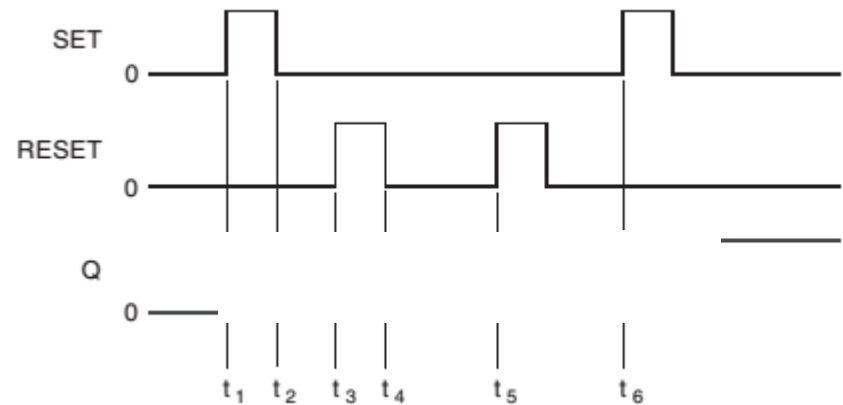
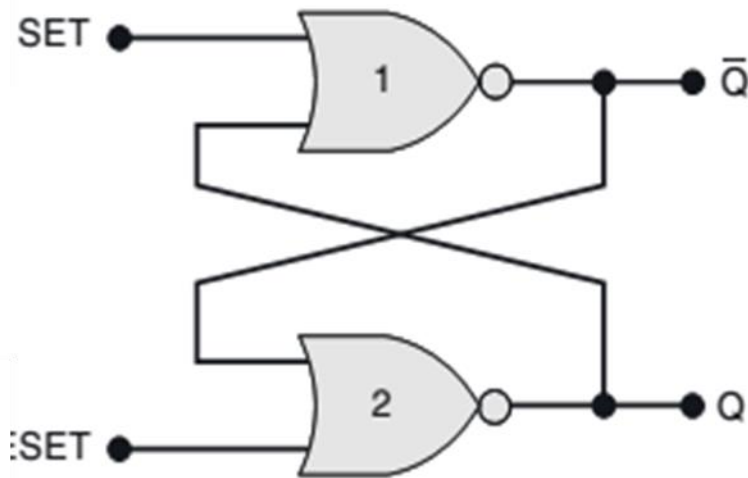
Latch com NOR

Resumo

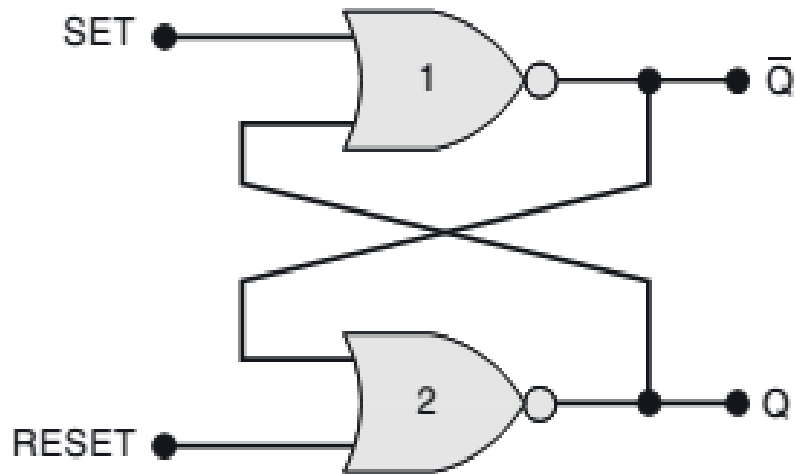
- SET=RESET=1
 - Tenta a mesmo tempo *setar* e *resetar* o latch
 - Produz $Q=Q'=0$
 - Se as entradas retornarem ao 0 simultaneamente o resultado é imprevisível
 - Condição inválida

Latch com NOR

- Considere inicialmente $Q = 0$ e determine a forma de onda da saída Q , para um latch NOR que tem as entradas mostradas na Figura



Latch com NOR



SET	RESET	Saída
0	0	Não muda
0	1	Q=0
1	0	Q=1
1	1	Inválida*

* Produz $Q=\bar{Q}=0$

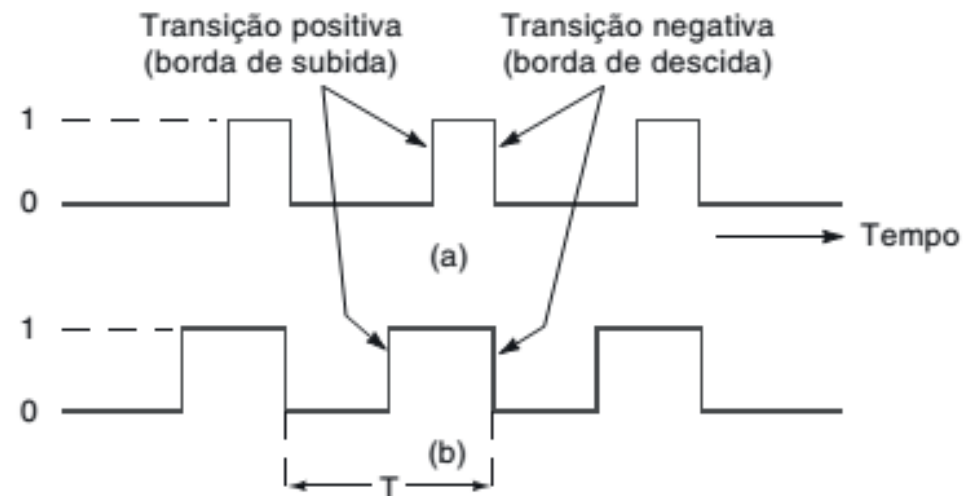
NOR *Gate Latch*

Resumo

- **SET = 0, RESET = 0** → Estado normal mantido. Não tem efeito na saída
- **SET = 1, RESET = 0** → Define o valor na saída $Q = 1$, que se mantém mesmo após SET retornar a 0.
- **SET = 0, RESET = 1** → Limpa o valor da saída $Q = 0$, que se mantém mesmo após RESET retornar a 0.
- **SET = 1, RESET = 1** → Tentativa de "setar" e "resetar" o latch ao mesmo tempo e produz $Q = \bar{Q} = 0$
 - A saída é imprevisível e essa condição não deve ser usada

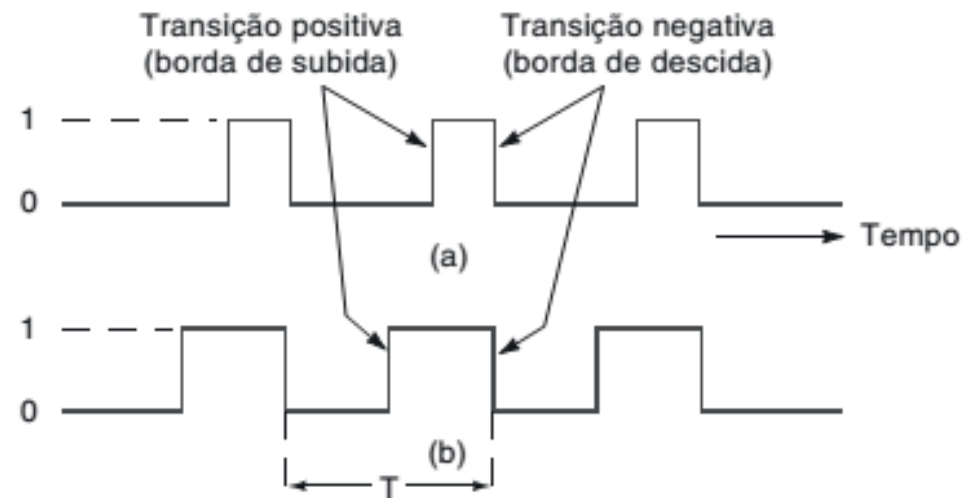
Sinal de Clock

- Sistemas síncronos
 - As saídas dos circuitos tem um momento exato para mudar de estado e este é determinado por um sinal de clock.
 - Sinal de clock é um trem de pulsos retangulares (onda quadrada).
 - Sinal de clock é distribuído para todo o sistema (sistema trabalha de forma sincronizada).



Sinal de Clock

- Sistemas síncronos
 - As transições (também denominadas bordas).
 - Quando o clock muda de 0 para 1, denomina-se transição positiva (borda de subida);
 - Quando muda de 1 para 0, denomina-se transição negativa (borda de descida).

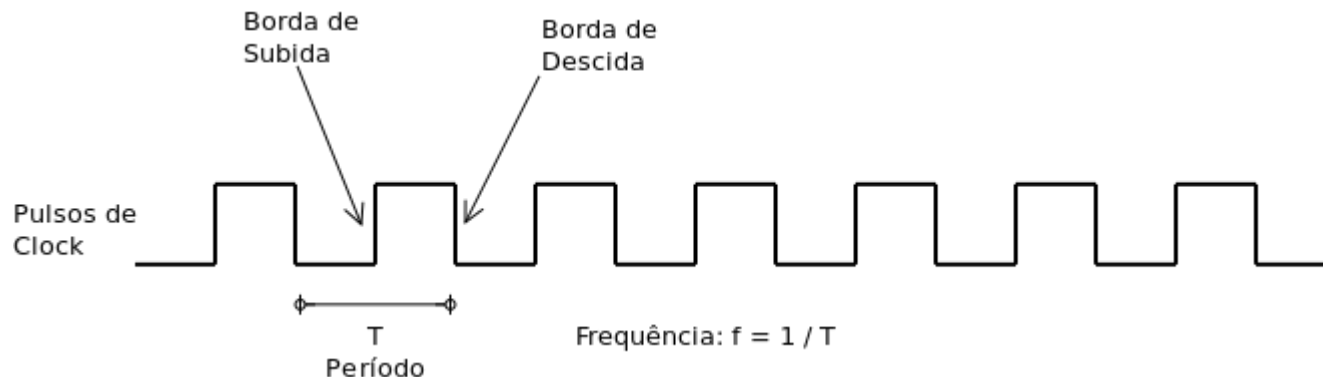


Sinal de Clock

- Sistemas assíncronos
 - Saída pode mudar de estado a qualquer momento em que uma ou mais entradas mudarem de estado.
 - A saída varia de acordo com as entradas no instante em que elas são aplicadas.
 - Em sistemas assíncronos não existe um tempo determinado para que processo execute uma determinada tarefa.
 - Projeto e análise de defeitos são mais complicados.

Sistemas Síncronos

- Velocidade da operação depende da frequência do ciclo de clock (1Hz=1ciclo/segundo).
- O ciclo é medido desde uma borda de subida/descida até a próxima, este tempo (segundos) é chamado de período T.
- É possível sincronizar eventos usando **flip-flops com clock**
 - Projetados para só mudar de estado em uma das transições do sinal de clock.



Clock Interno

- Em um computador, todas as atividades necessitam de sincronização.
- O **clock interno** (ou **clock**), basicamente, atua como um sinal para sincronismo.
- Quando os dispositivos do computador recebem o sinal de executar suas atividades, dá-se a esse acontecimento o nome de "pulso de clock".
- A medição do clock é feita em *Hertz* (Hz), a unidade padrão de medidas de frequência.
- Se um processador trabalha à 800 Hz, por exemplo, significa que ele é capaz de lidar com 800 operações de ciclos de clock por segundo

Resumo:

Sinais de Relógio (*Clock*) e Flip-Flops Síncronos

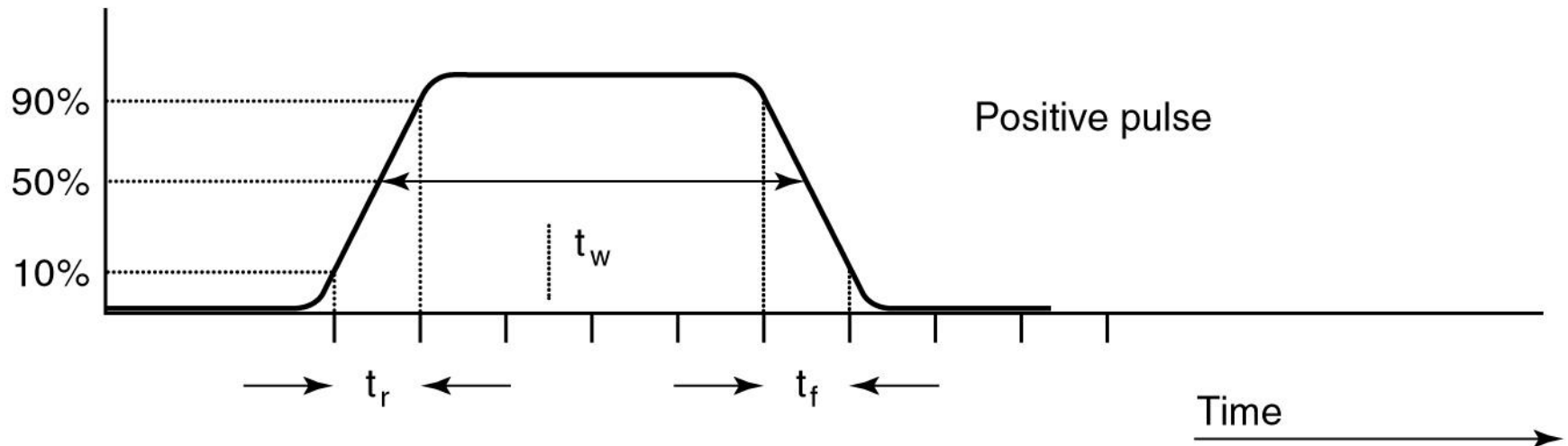
- Sistemas digitais podem operar tanto assincronamente como sincronamente:
 - Sistema assíncrono → saídas podem mudar a qualquer tempo em que a(s) entrada(s) mude(m).
 - Sistema síncrono → saídas podem mudar somente em tempos específicos, quando há um ciclo de *clock*.

Flip-flop com Clock

- Características:
 - Entradas de controle síncronas
 - Determina a saída do dispositivo (FF).
 - Entrada de clock é denominada CLK, CK ou CP
 - Determina QUANDO as saídas serão alteradas.
 - Na maioria dos FFs com clock, a entrada *CLK* é **disparada por borda**, o que significa que essa entrada é ativada pela transição do sinal de clock.
 - A entrada de clock é indicado por um pequeno triângulo na entrada *CLK*.

Pulsos Digitais

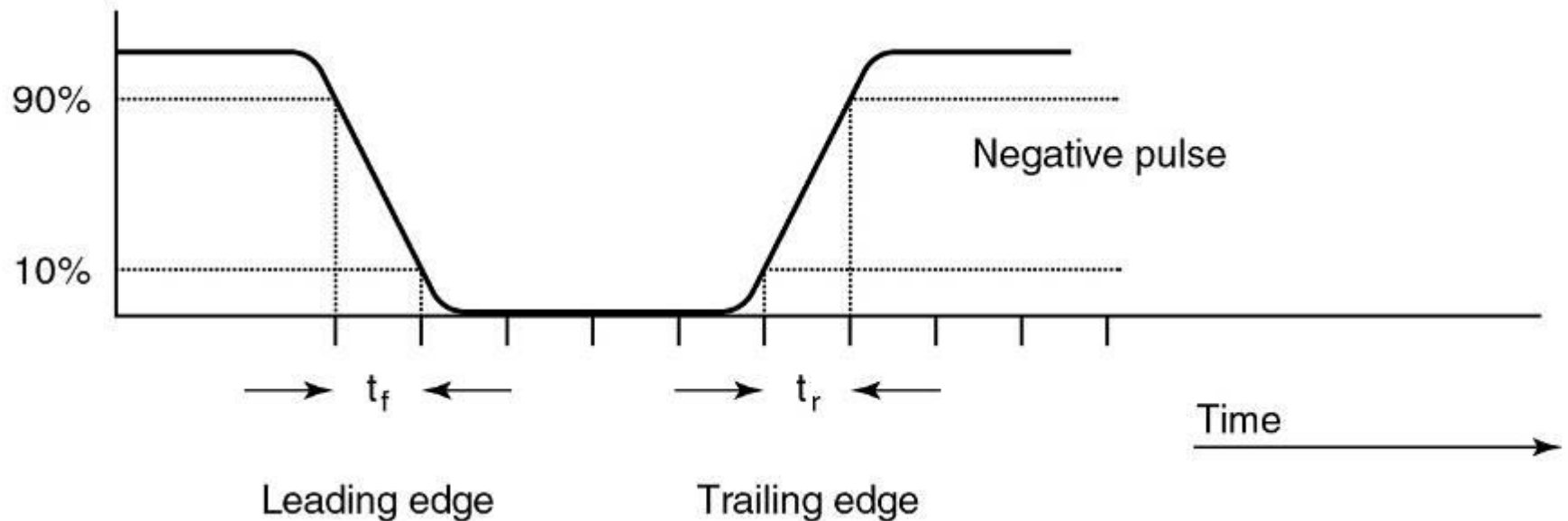
Sinais que trocam entre os estados baixo e alto são chamados de forma de onda



Um pulso positivo tem um nível ativo-ALTO

Pulsos Digitais

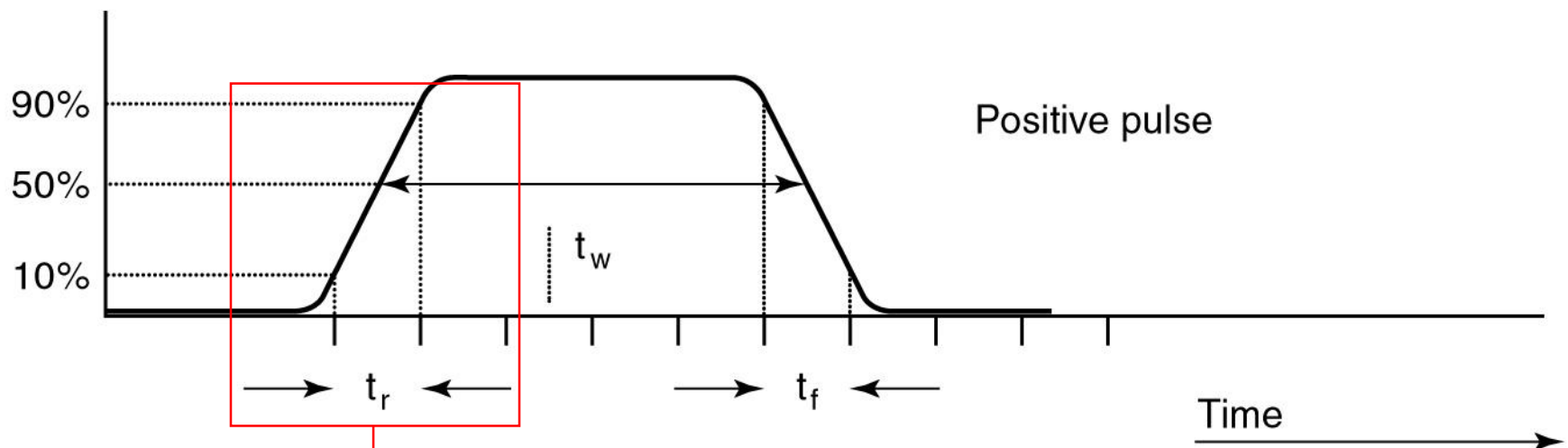
Sinais que trocam entre os estados baixo e alto são chamados de forma de onda



**Um pulso negativo tem
um nível ativo-BAIXO**

Pulsos Digitais

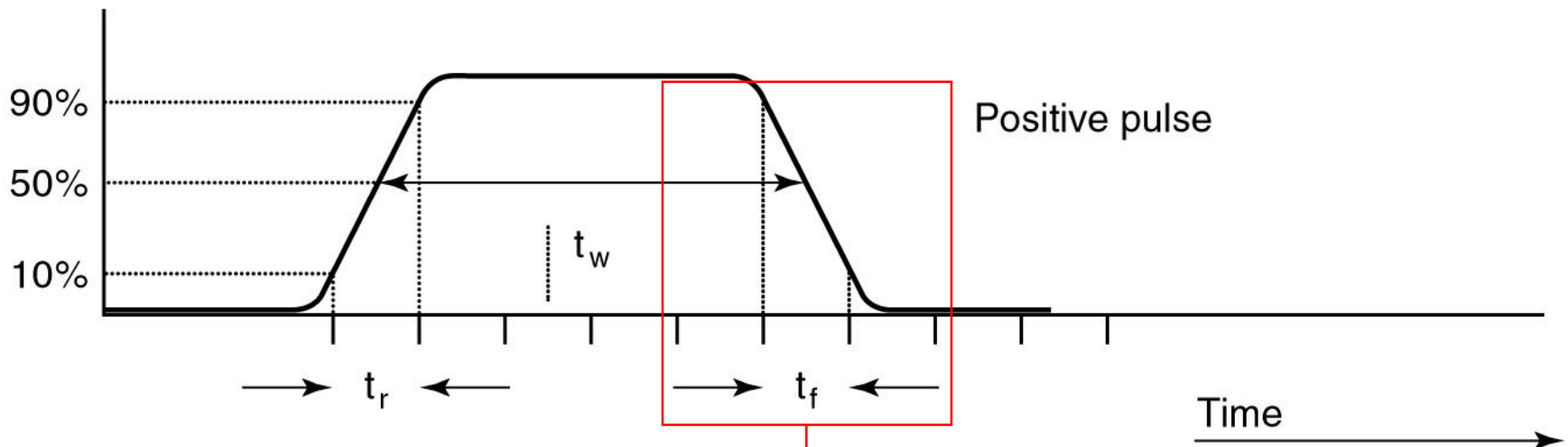
- Em um circuito real, há um tempo para o pulso de uma forma de onda mudar de um nível para outro
 - A transição de um BAIXO para ALTO em um pulso positivo é chamada de *rise time* (t_r) ou tempo de subida



**Medido entre os pontos de 10% e 90%
do lado de subida da forma de onda de tensão**

Pulsos Digitais

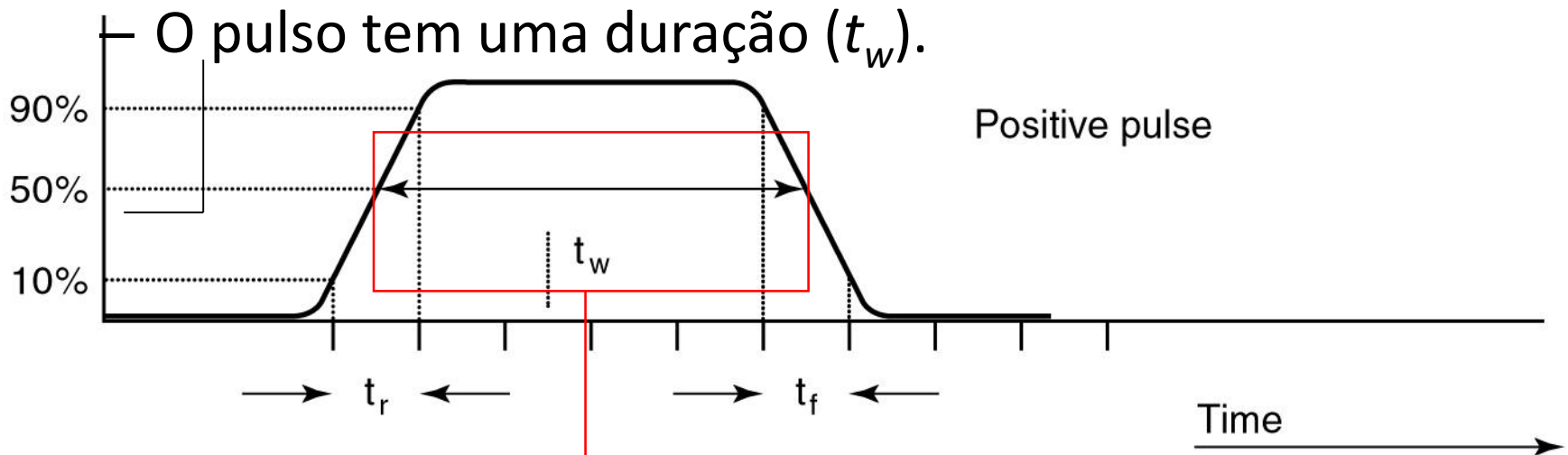
- Em um circuito real, há um tempo para o pulso de uma forma de onda mudar de um nível para outro
 - A transição de um ALTO para BAIXO em um pulso positivo é chamada de *fall time* (t_f) ou tempo de “queda”



Medido entre os pontos 90% e 10% do lado de descida da forma de onda de tensão

Pulsos Digitais

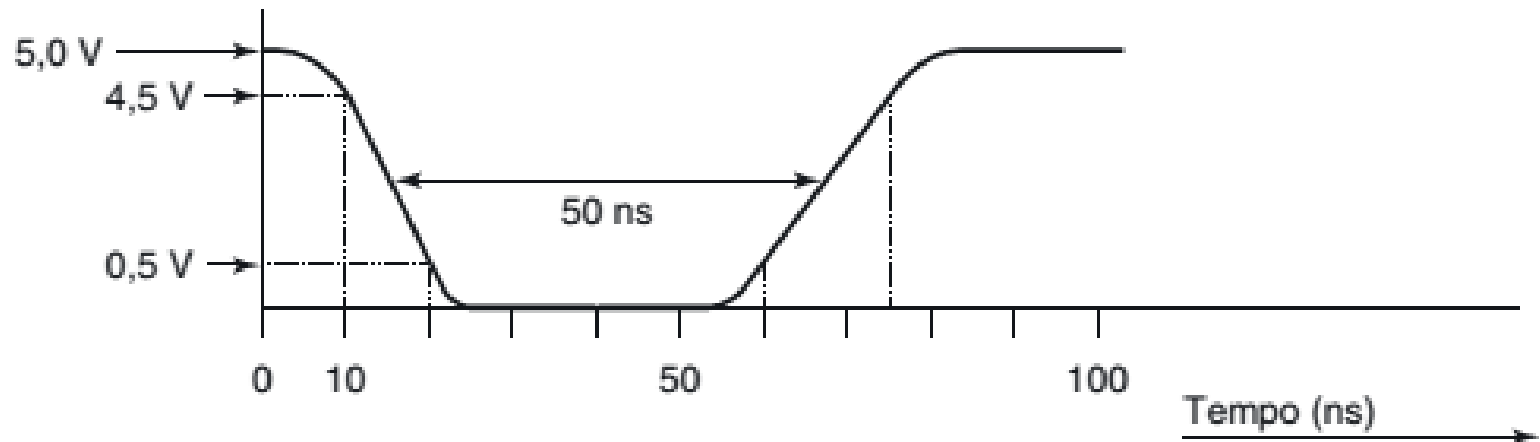
- Em um circuito real, há um tempo para o pulso de uma forma de onda mudar de um nível para outro



Tempo entre os pontos quando a subida e a descida estão a 50% do nível de tensão ALTO

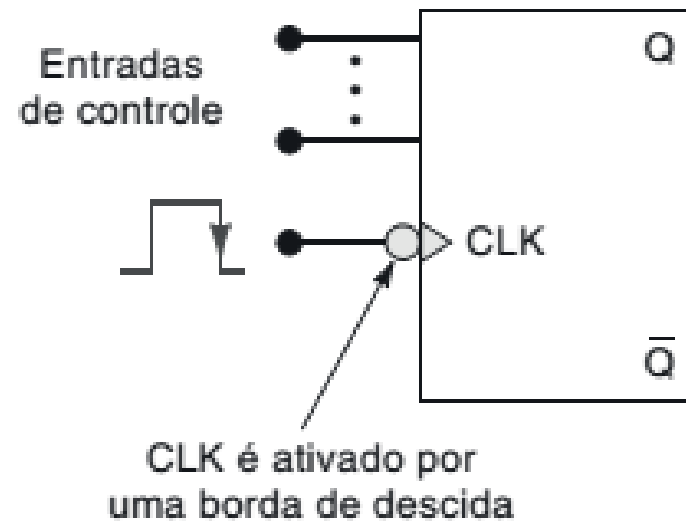
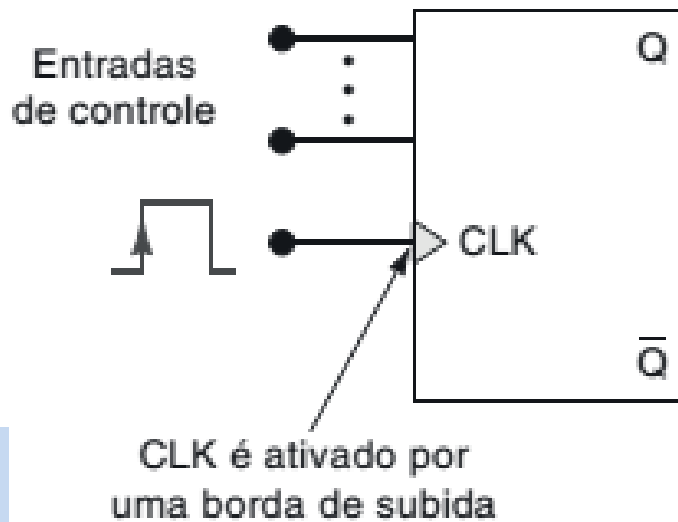
Exemplo

- Quando um microcontrolador quer ter acesso a dados em sua memória externa, ativa um pino de saída em estado ativo-BAIXO chamado \overline{RD} (read). O pulso \overline{RD} possui as seguintes características: largura t_w de 50 ns, tempo de subida t_r de 15 ns e tempo de descida t_f de 10 ns. O pulso varia entre 0V e 5V. Desenhe o pulso \overline{RD} em escala.



Flip-flop com Clock

- Características:
 - Entrada de *clock* é disparada por borda de subida ou descida.
 - O FF pode ter mais de uma entrada, mas a saída depende do sinal de *clock*, fazendo com que o sistema seja sincronizado.

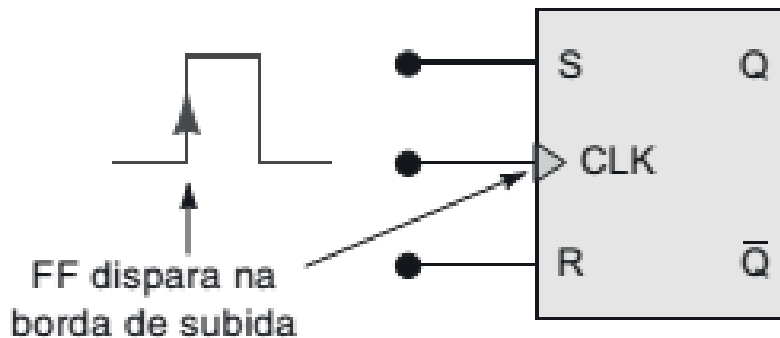


Flip-flop SR com Clock

- Flip-flop disparado na borda de subida do sinal de clock.
- As entradas S e R controlam o estado do FF, como descrito anteriormente para um latch de NOR.
- A entrada CLK é a entrada de disparo (**trigger**) que provoca a mudança de estado de acordo com as entradas S e R .
- FF não responde a essas entradas até que ocorra uma borda de subida no sinal de clock.

Set	Reset	Output
0	0	No change
1	0	$Q = 1$
0	1	$Q = 0$
1	1	Invalid*

*Produces $Q = \bar{Q} = 0$.



Entradas

Saída

S	R	CLK	Q
0	0	↑	Q_0 (não muda)
1	0	↑	1
0	1	↑	0
1	1	↑	Ambíguo

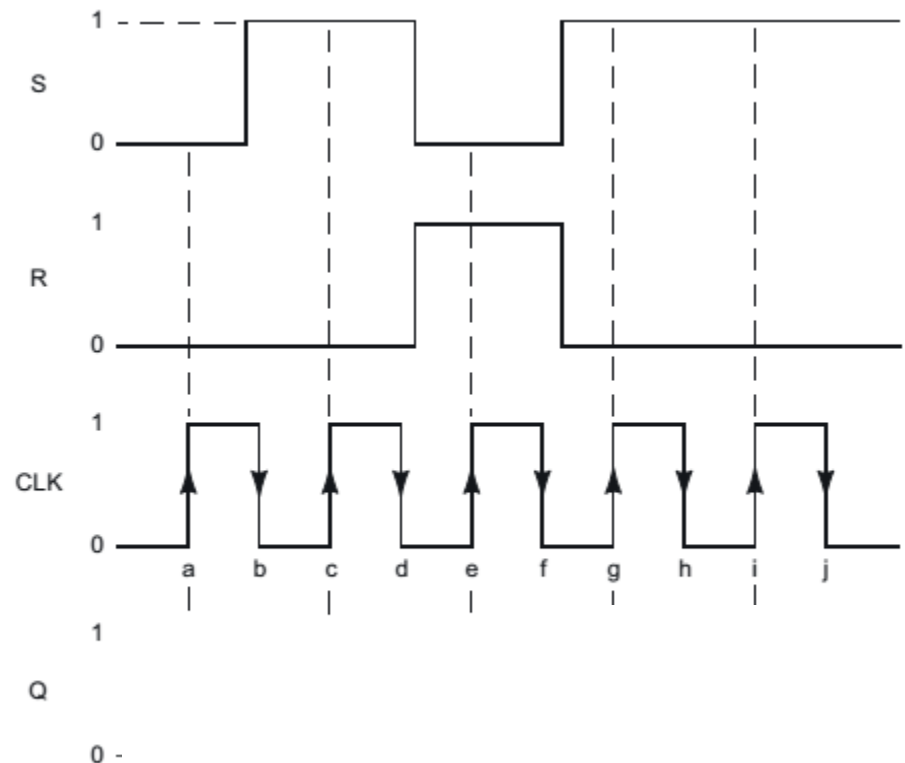
Q_0 é o nível de saída anterior a ↑ de CLK.
↓ de CLK não produz mudança em Q.

Flip-flop SR com Clock

Entradas			Saída
S	R	CLK	Q
0	0	↑	Q_0 (não muda)
1	0	↑	1
0	1	↑	0
1	1	↑	Ambíguo

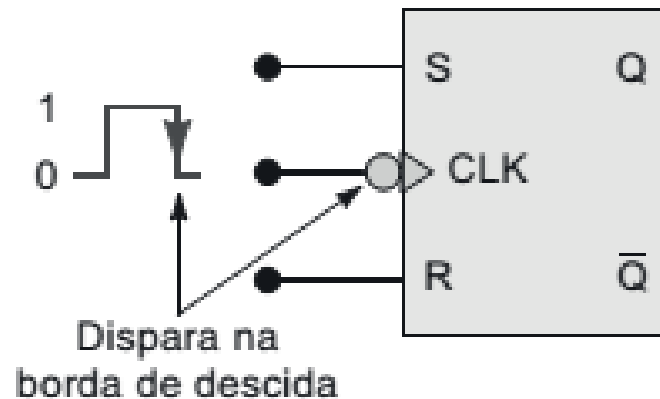
Q_0 é o nível de saída anterior a ↑ de CLK.
↓ de CLK não produz mudança em Q.

Formas de onda da operação de um flip-flop S-R com clock, disparado pela borda positiva de um pulso de clock



Flip-flop SR com Clock

- *Flip-Flop S-R* disparado na borda de descida que ocorre na entrada *CLK*

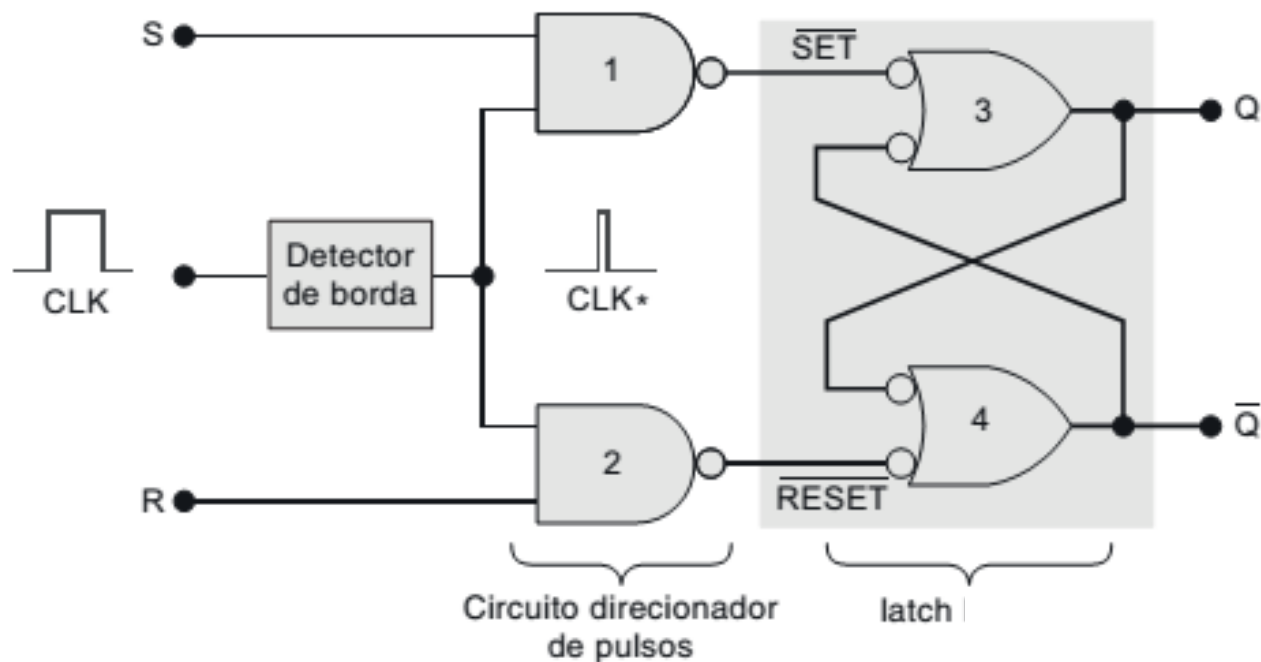


Entradas			Saída
S	R	CLK	Q
0	0	↓	Q_0 (não muda)
1	0	↓	1
0	1	↓	0
1	1	↓	Ambíguo

Flip-flop disparado por borda

Circuito Interno

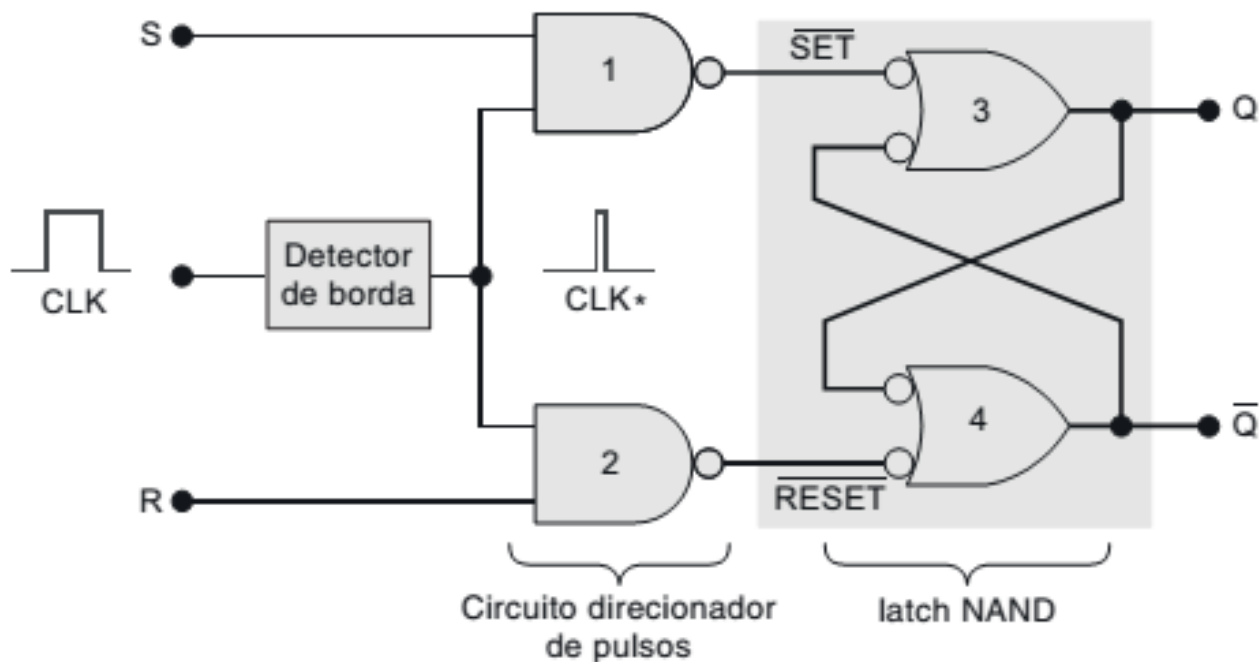
- Circuito interno dividido em 3 partes
 - Latch NAND ou NOR?
 - Circuito direcionador/encaminhador de pulsos
 - Circuito detector de borda



Flip-flop disparado por borda

Circuito Interno

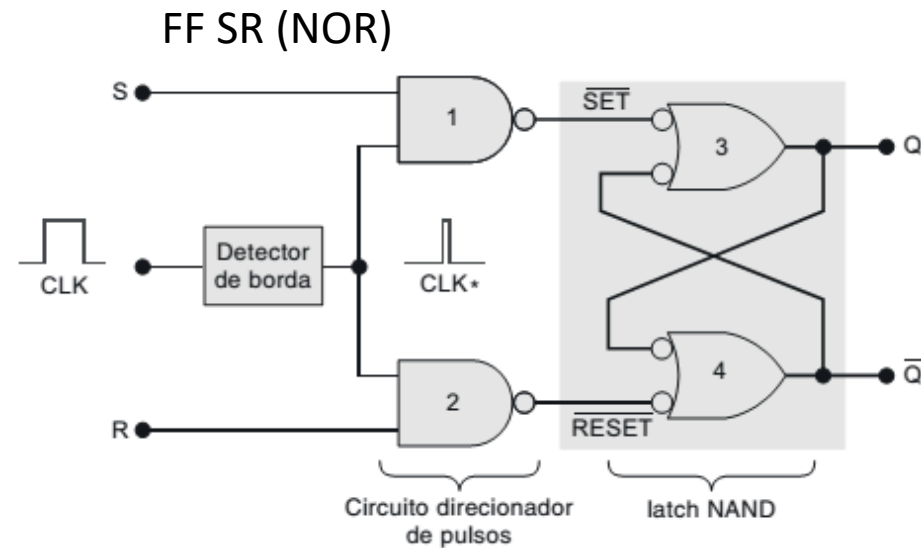
- Circuito interno dividido em 3 partes
 - Latch NAND: 3,4
 - Circuito direcionador/encaminhador de pulsos: 1,2
 - Circuito detector de borda



Flip-Flop Disparado por Borda

Circuito Interno

- O circuito detector de borda produz um pulso estreito e positivo (CLK^*), que ocorre no instante da transição ativa do pulso na entrada CLK .

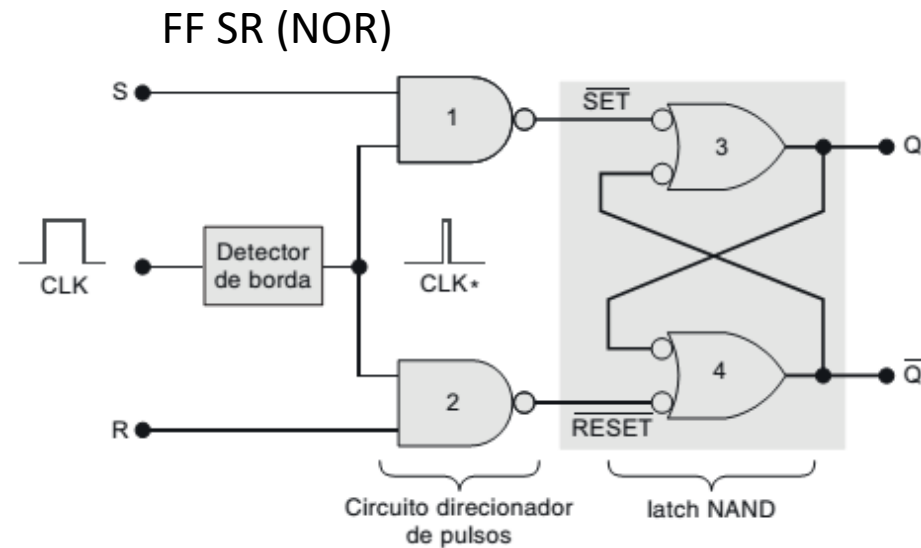


NAND			LATCH NAND		
A	B	S	SET	RESET	Saída
0	0	1	0	0	Inválida*
0	1	1	0	1	$Q=1$
1	0	1	1	0	$Q=0$
1	1	0	1	1	Não muda

Flip-Flop Disparado por Borda

Circuito Interno

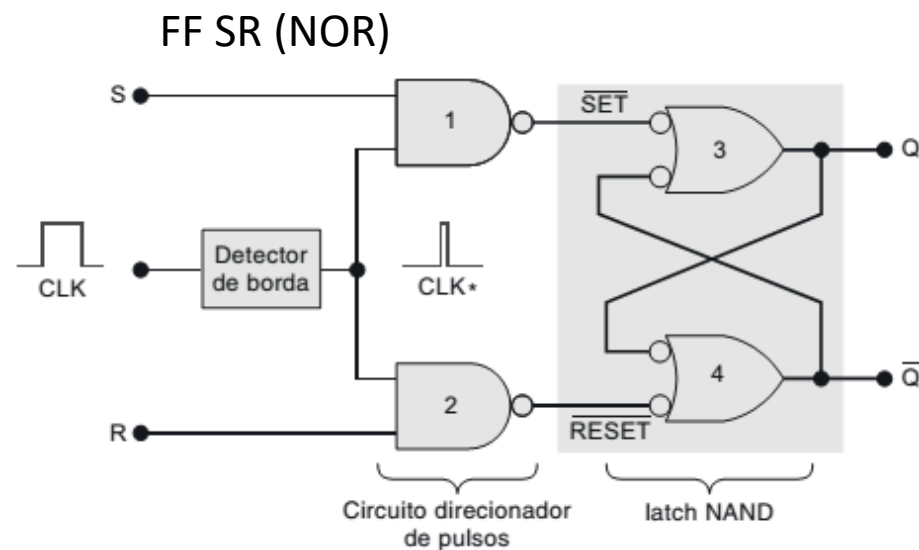
- O circuito direcionador de pulsos 'direciona' esse pulso estreito para a entrada *SET* ou a *RESET* do latch.



NAND			LATCH NAND		
A	B	S	SET	RESET	Saída
0	0	1	0	0	Inválida*
0	1	1	0	1	Q=1
1	0	1	1	0	Q=0
1	1	0	1	1	Não muda

Flip-Flop Disparado por Borda

Circuito Interno



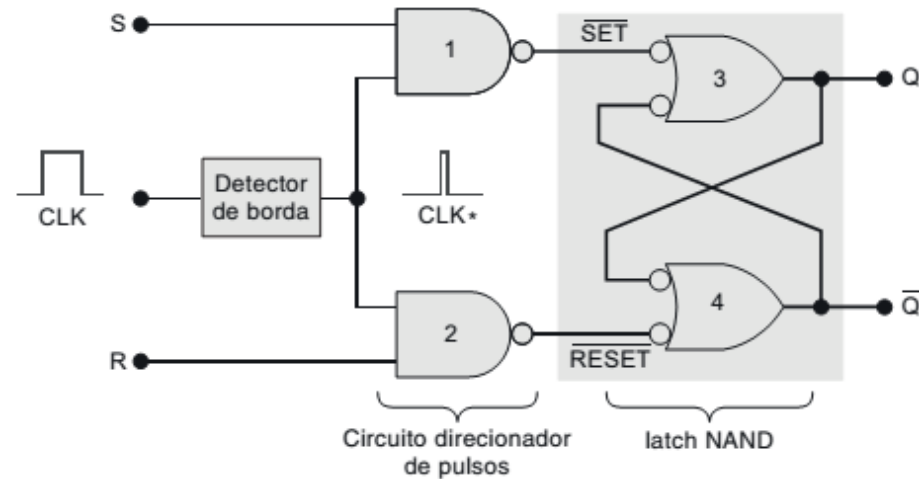
- Com $S = 1$ e $R = 0$, o sinal CLK^* é invertido na passagem pela NAND no 1, e produz um pulso de nível BAIXO na entrada SET, o qual resulta em $Q = 1$.

NAND			LATCH NAND		
A	B	S	SET	RESET	Saída
0	0	1	0	0	Inválida*
0	1	1	0	1	Q=1
1	0	1	1	0	Q=0
1	1	0	1	1	Não muda

Flip-flop disparado por borda

Circuito Interno

- Com $S = 0$ e $R = 1$, o sinal CLK^* é invertido na passagem pela NAND no 2, e produz um pulso de nível baixo na entrada RESET do latch, o qual resulta em $Q = 0$



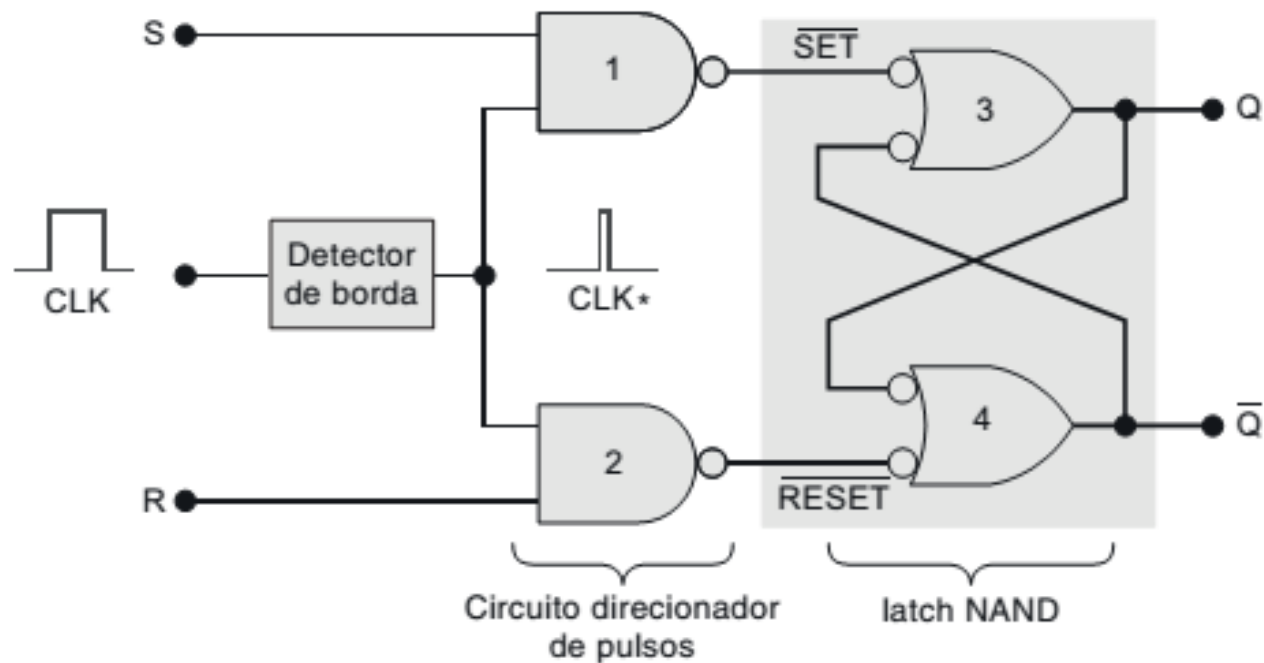
NAND

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0

LATCH NAND

SET	RESET	Saída
0	0	Inválida*
0	1	Q=1
1	0	Q=0
1	1	Não muda

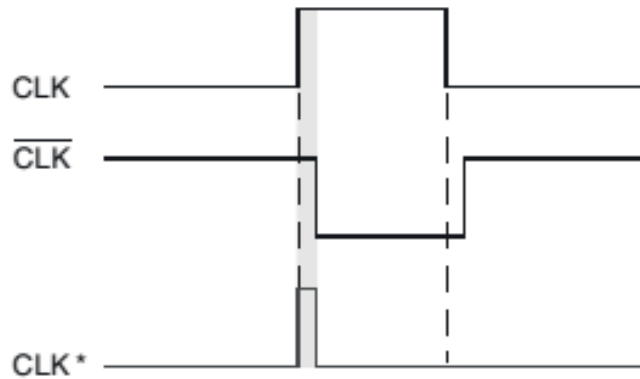
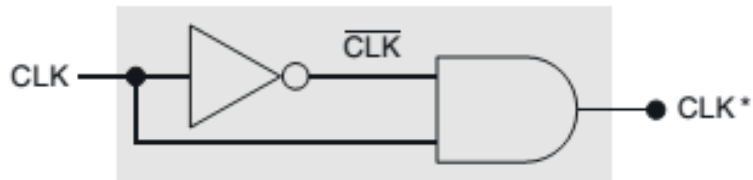
Flip-flop disparado por borda



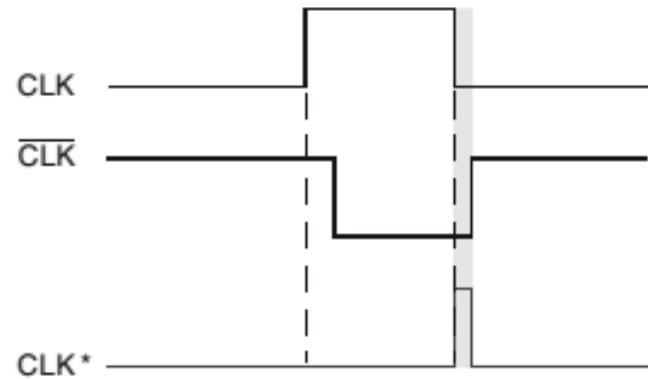
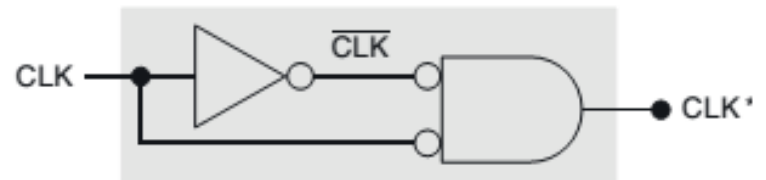
Detector de borda

- Leva em consideração atraso de resposta das portas lógicas (nanossegundos) de forma a produzir um pulso estreito (*spike*) durante as bordas.
- As saída Q é afetada por um curto período de tempo após a ocorrência da borda ativa.

Detector de borda



Circuito Detector de Borda Positiva



Circuito Detector de Borda Negativa

SN54279

QUADRUPLE S-R LATCHES

FUNCTION TABLE
(each latch)

INPUTS		OUTPUT Q
\bar{S}^\dagger	\bar{R}	
H	H	Q_0
L	H	H
H	L	L
L	L	H^\ddagger

H = high level L = low level

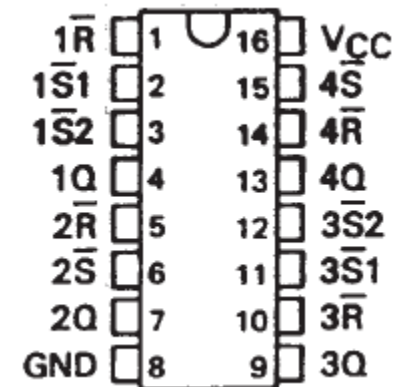
† For latches with double S inputs:

Q_0 = the level of Q before the indicated input conditions were established.

‡ This configuration is nonstable: that is, it may not persist when the \bar{S} and \bar{R} inputs return to their inactive (high) level.

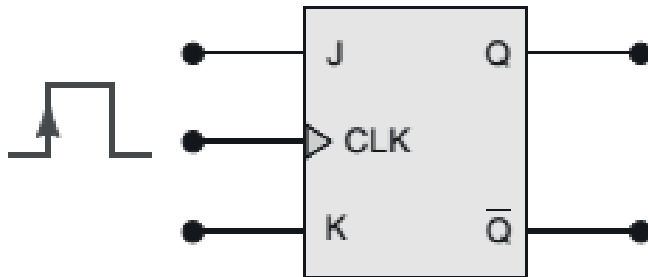
H = both \bar{S} inputs high

L = one or both \bar{S} inputs low



Flip Flop J-K com Clock

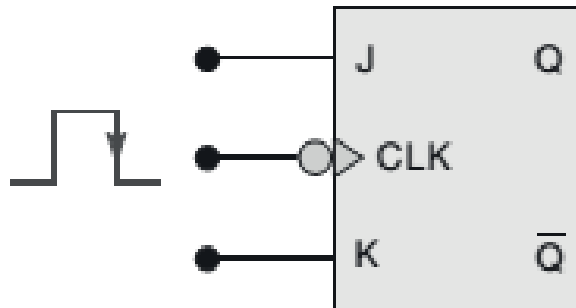
- Funciona da mesma maneira que o FF SR com clock, mas sem a condição de ambiguidade $J=K=1$.
- Para $J=K=1$; o FF sempre muda para o **estado lógico oposto** no instante da **borda de subida do sinal de clock**. Modo comutação (toggle mode)



J	K	CLK	Q
0	0	↑	Q_0 (não muda)
1	0	↑	1
0	1	↑	0
1	1	↑	\bar{Q}_0 (comuta)

Flip Flop J-K com Clock

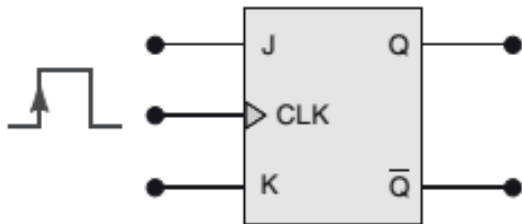
- Flip Flop J-K com clock disparado por bordas de descida.



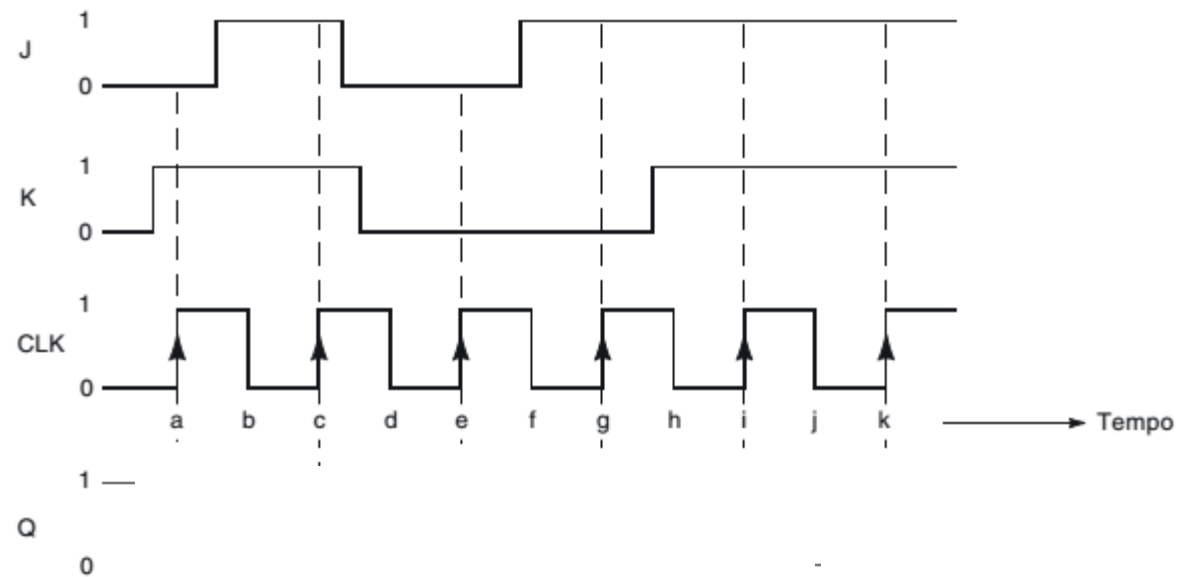
J	K	CLK	Q
0	0	↓	Q_0 (não muda)
1	0	↓	1
0	1	↓	0
1	1	↓	$\overline{Q_0}$ (comuta)

- A condição $J = K = 1$, que gera a operação de comutação da saída, é bastante utilizada em todos os tipos de contadores binários.

Flip Flop J-K com Clock



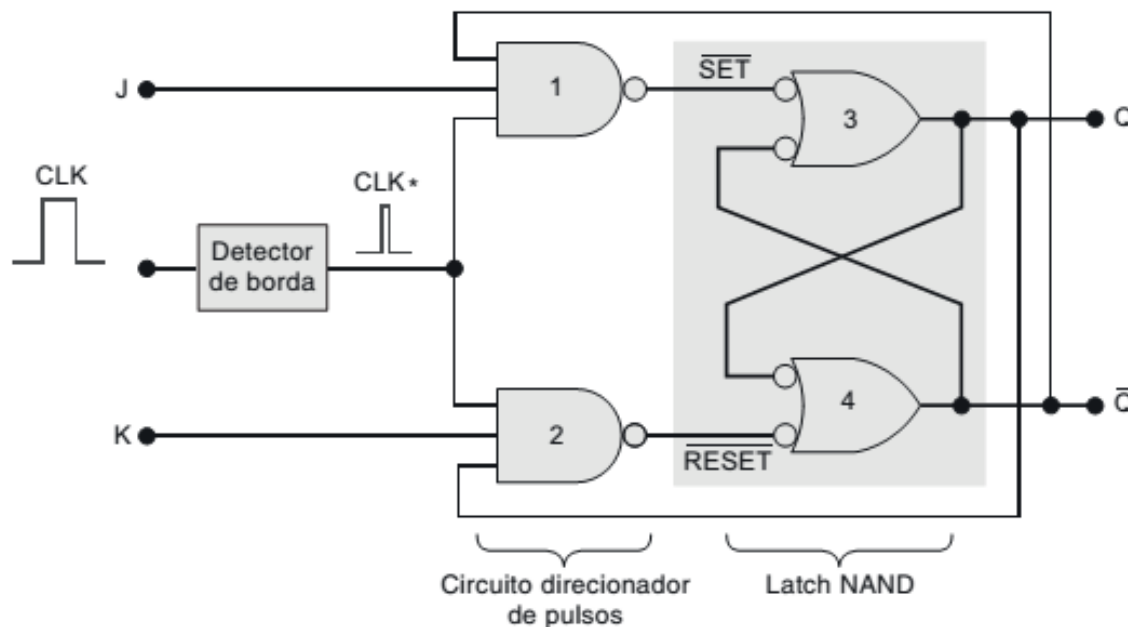
J	K	CLK	Q
0	0	\uparrow	Q_0 (não muda)
1	0	\uparrow	1
0	1	\uparrow	0
1	1	\uparrow	\bar{Q}_0 (comuta)



Circuito Interno de um Flip-Flop J-K

Disparado por Borda

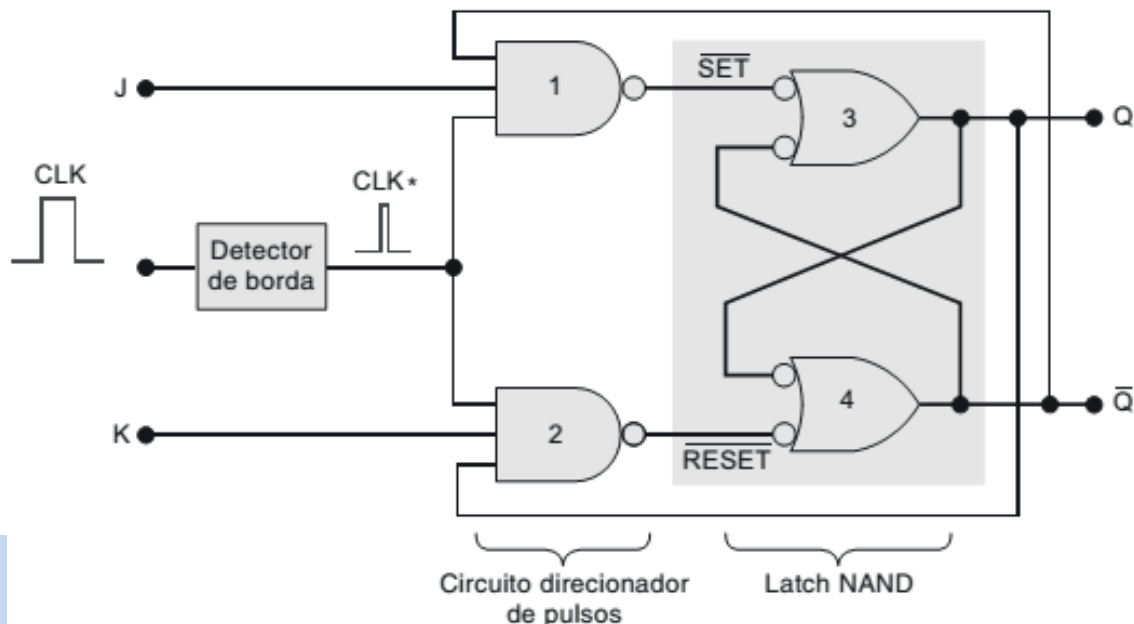
- A conexão de realimentação é quem confere ao flip-flop J - K a operação de comutação para a condição em que $J = K = 1$.



J	K	CLK	Q
0	0	↑	Q_0 (não muda)
1	0	↑	1
0	1	↑	0
1	1	↑	\bar{Q}_0 (comuta)

Circuito Interno de um Flip-Flop J-K Disparado por Borda

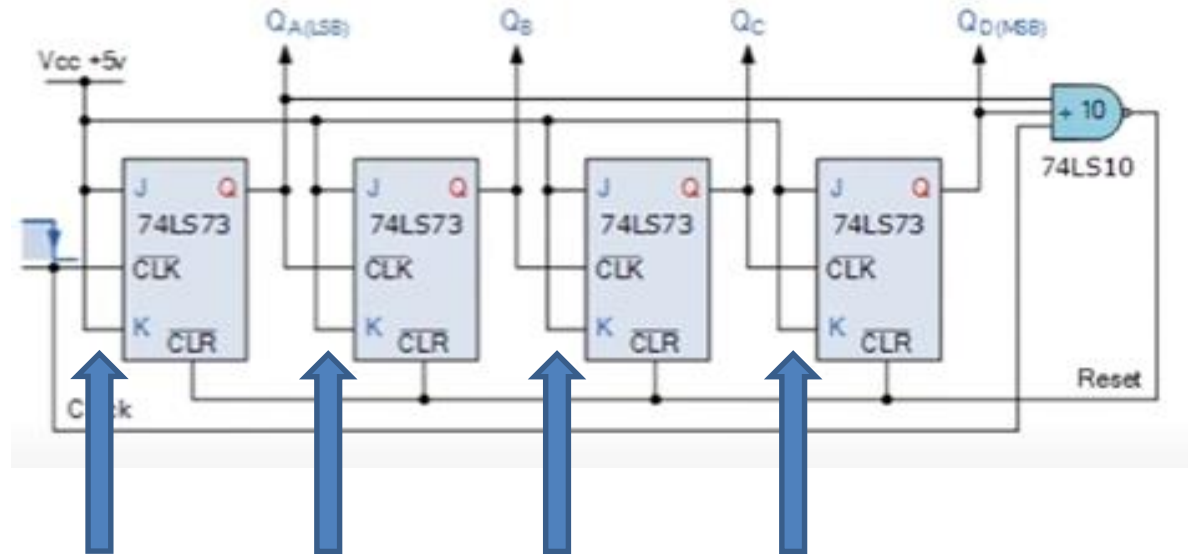
Para que a operação de comutação funcione conforme descrito, o pulso CLK^* tem de ser muito estreito. Ele tem de retornar para o nível 0 antes que as saídas Q e \bar{Q} comutem para seus novos valores; caso contrário, os novos valores de Q e \bar{Q} farão com que o pulso CLK^* comute a saída do latch novamente.



J	K	CLK	Q
0	0	↑	Q_0 (não muda)
1	0	↑	1
0	1	↑	0
1	1	↑	\bar{Q}_0 (comuta)

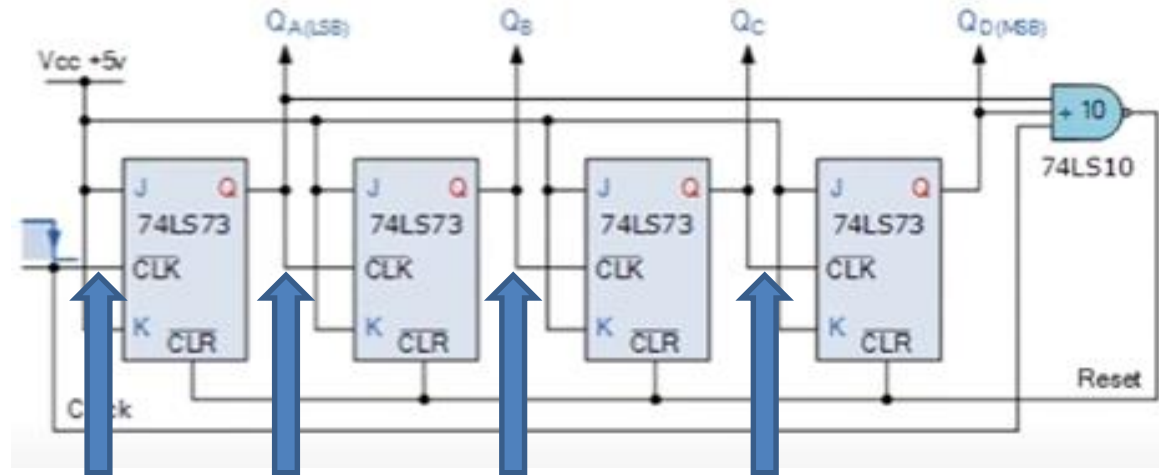
Contador decimal feito com Flip-Flop J K

QD	QC	QB	QA	DEC
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9



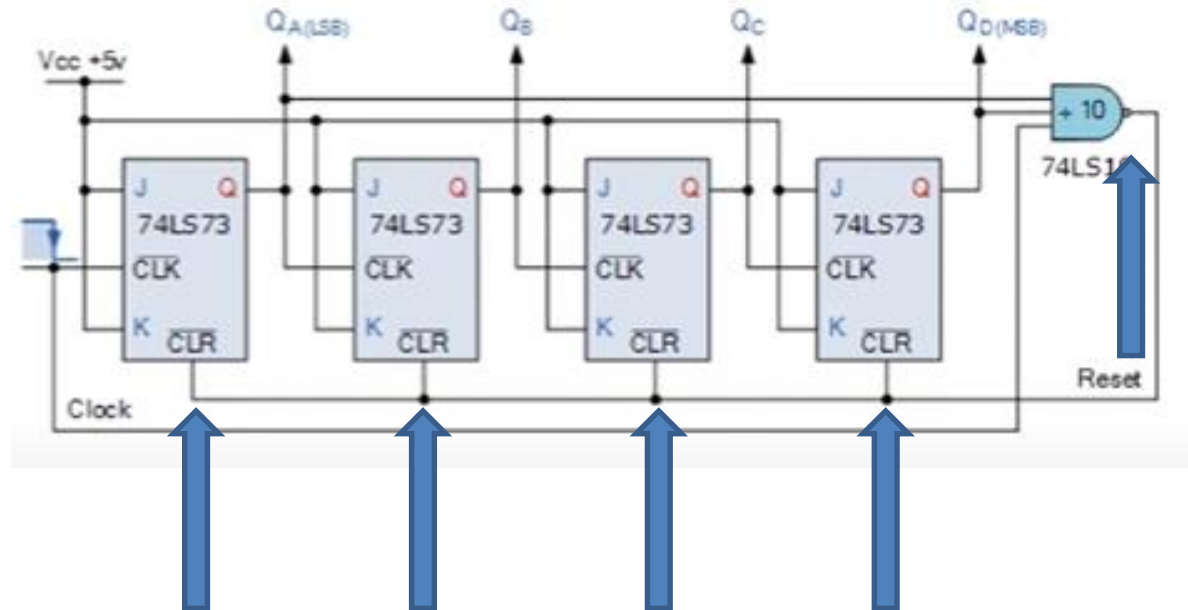
Contador decimal feito com Flip-Flop J K

QD	QC	QB	QA	DEC
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9



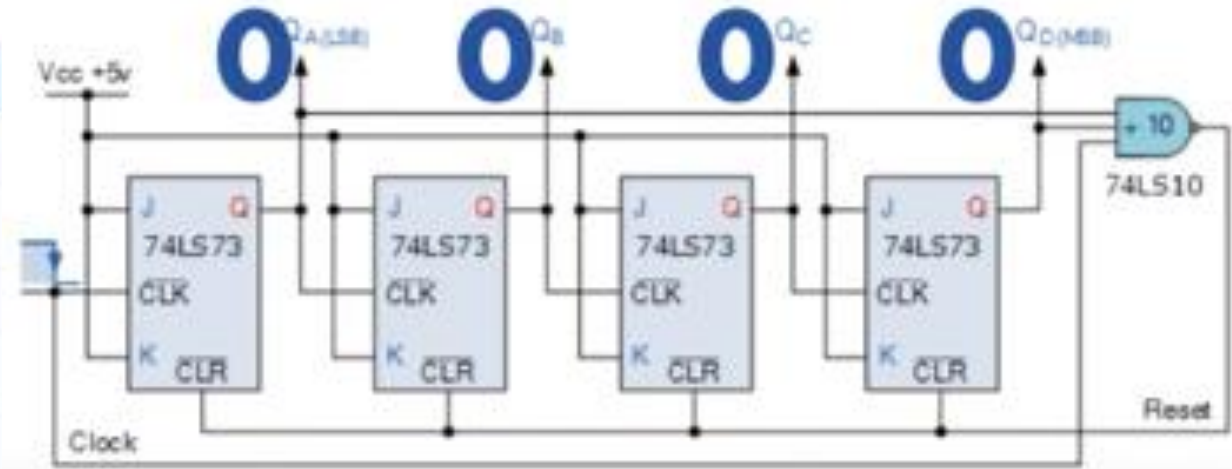
Contador decimal feito com Flip-Flop J K

QD	QC	QB	QA	DEC
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9



Contador decimal feito com Flip-Flop J K

QD	QC	QB	QA	DEC
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

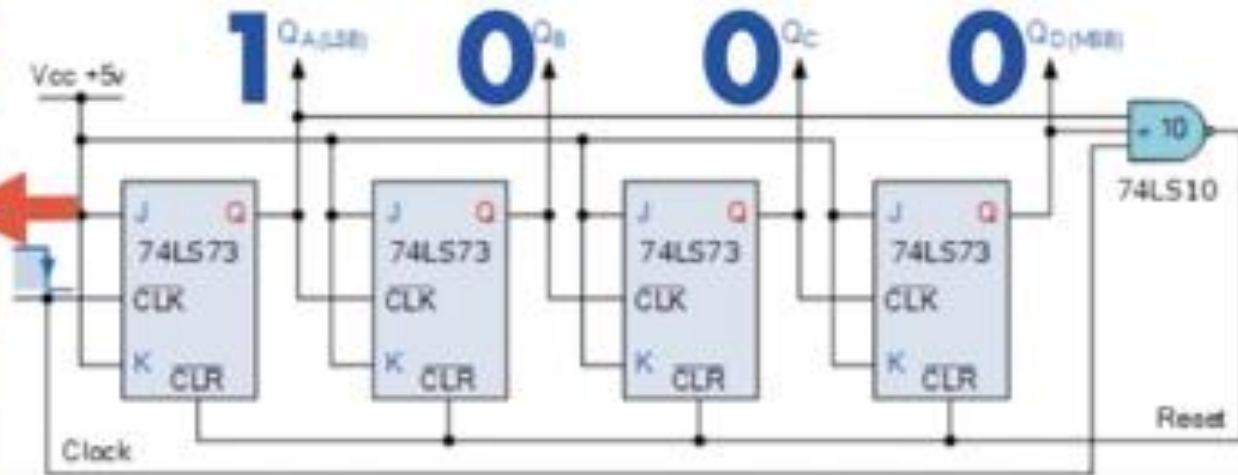


J	K	CLK	Q	Q\
0	0	↓	Q0	Q0\
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	Toggle	

Toggle = Novo estado será o anterior invertido.
Se estava 0 fica 1, se estava 1 fica 0

Contador decimal feito com Flip-Flop J K

QD	QC	QB	QA	DEC
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

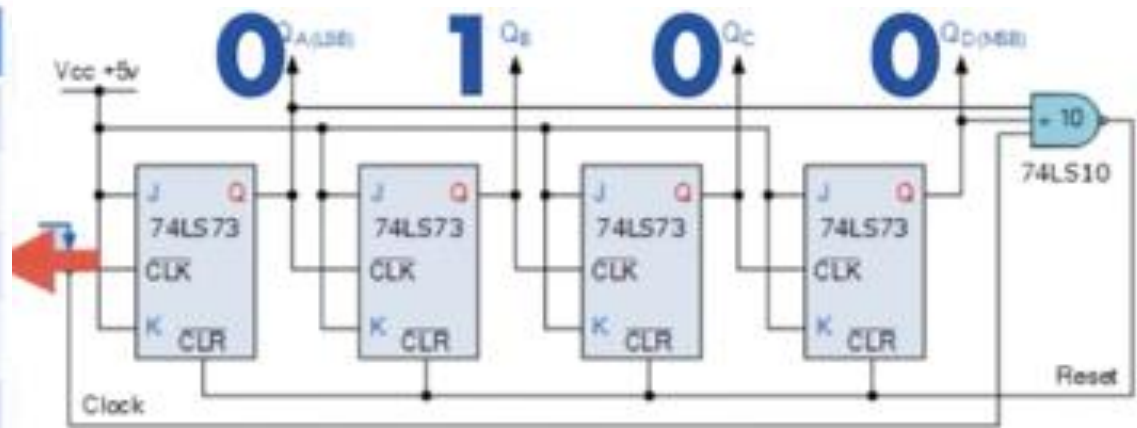


J	K	CLK	Q	Q\
0	0	↓	Q0	Q0\
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	Toggle	

Toggle = Novo estado será o anterior invertido.
Se estava 0 fica 1, se estava 1 fica 0

Contador decimal feito com Flip-Flop J K

QD	QC	QB	QA	DEC
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

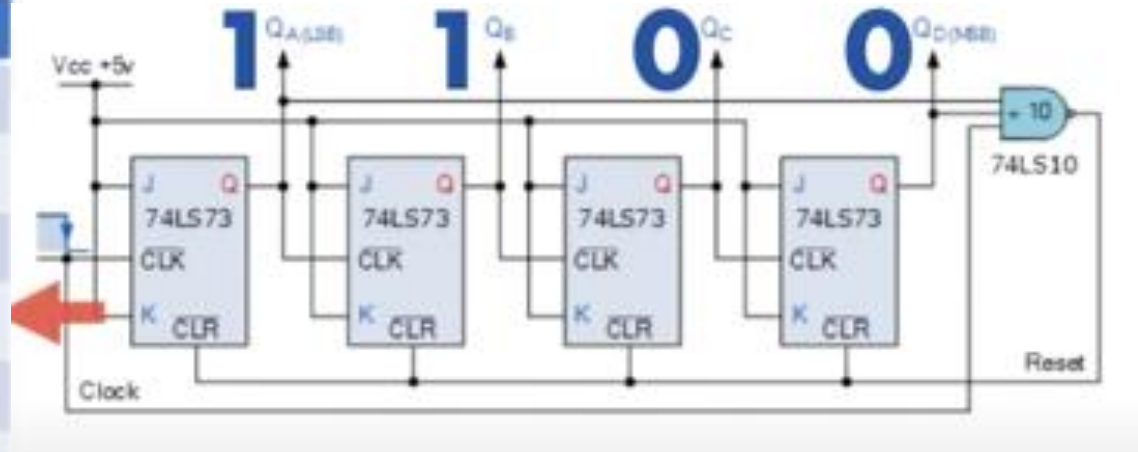


J	K	CLK	Q	Q\
0	0	↓	Q0	Q0\
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	Toggle	

Toggle = Novo estado será o anterior invertido.
Se estava 0 fica 1, se estava 1 fica 0

Contador decimal feito com Flip-Flop J K

QD	QC	QB	QA	DEC
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

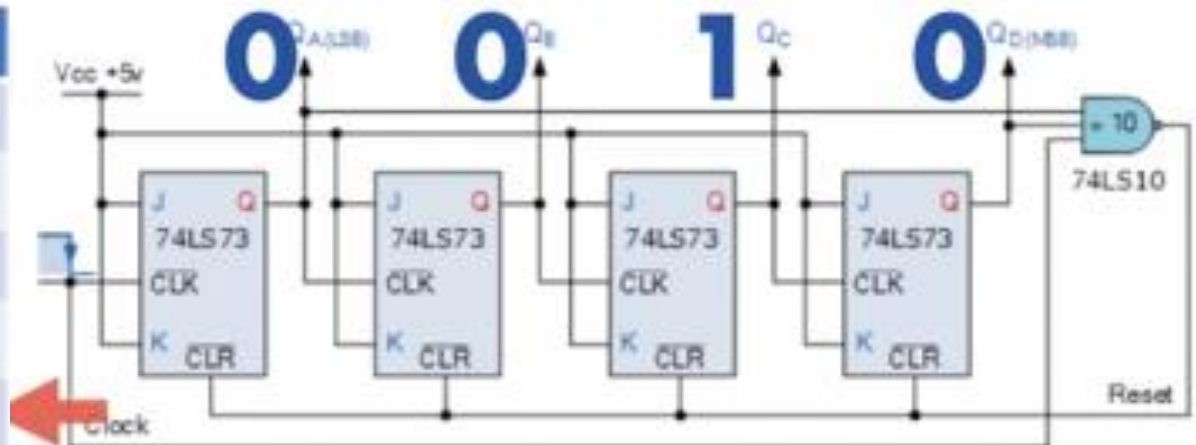


J	K	CLK	Q	Q\
0	0	↓	Q0	Q0\
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	Toggle	

Toggle = Novo estado será o anterior invertido.
Se estava 0 fica 1, se estava 1 fica 0

Contador decimal feito com Flip-Flop J K

QD	QC	QB	QA	DEC
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

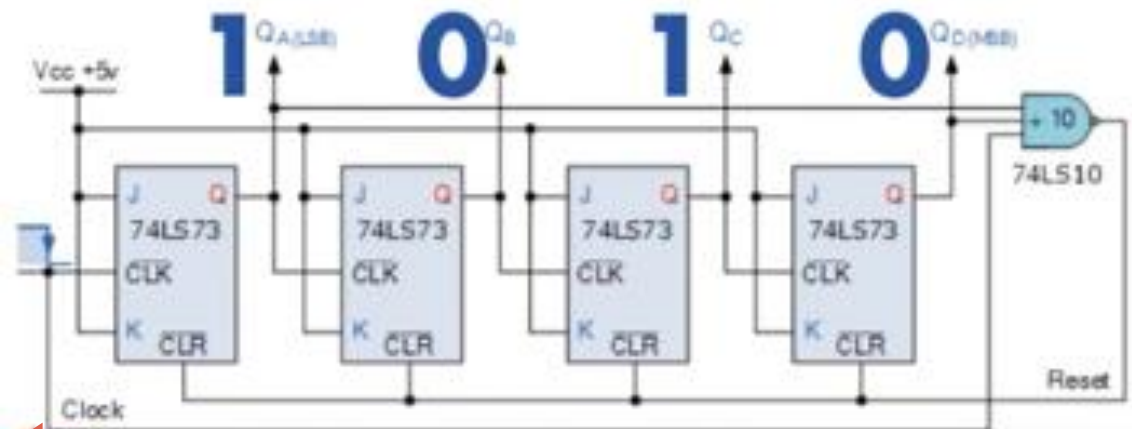


J	K	CLK	Q	Q\
0	0	↓	Q0	Q0\
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	Toggle	

Toggle = Novo estado será o anterior invertido.
Se estava 0 fica 1, se estava 1 fica 0

Contador decimal feito com Flip-Flop J K

QD	QC	QB	QA	DEC
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

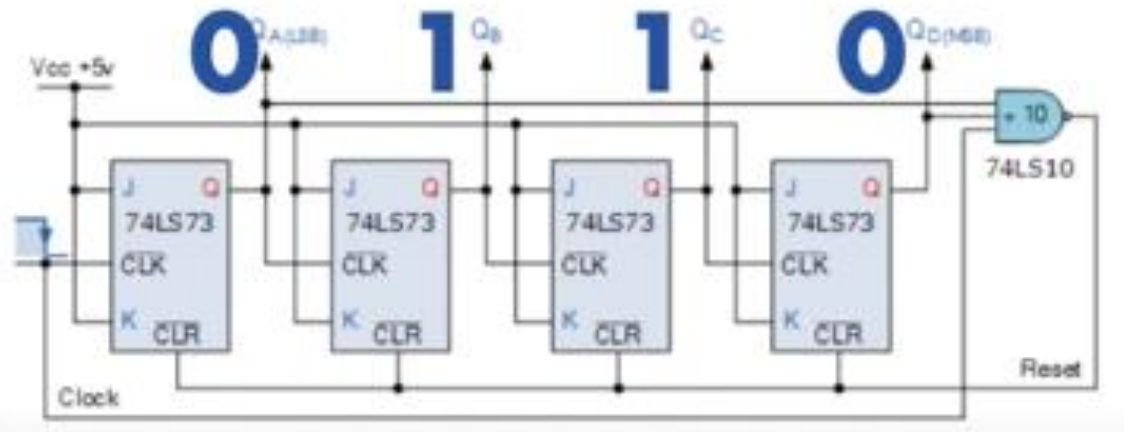


J	K	CLK	Q	Q\
0	0	↓	Q0	Q0\
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	Toggle	

Toggle = Novo estado será o anterior invertido.
Se estava 0 fica 1, se estava 1 fica 0

Contador decimal feito com Flip-Flop J K

QD	QC	QB	QA	DEC
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

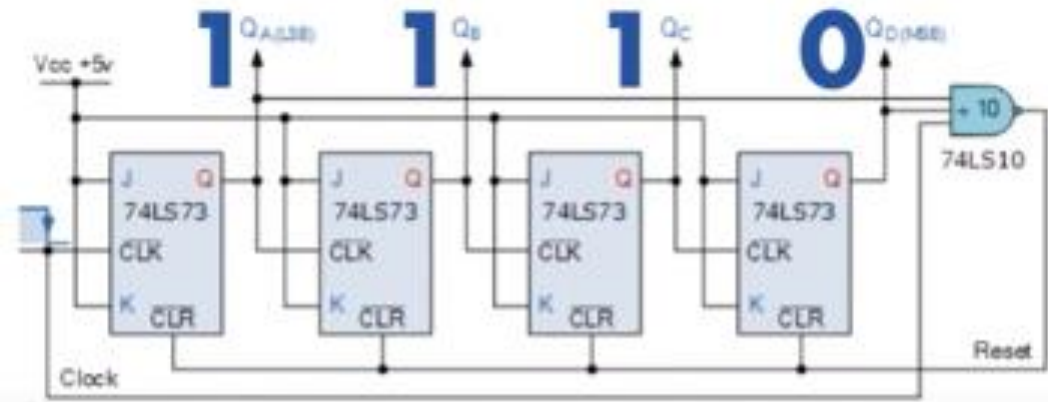


J	K	CLK	Q	Q\
0	0	↓	Q0	Q0\
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	Toggle	

Toggle = Novo estado será o anterior invertido.
Se estava 0 fica 1, se estava 1 fica 0

Contador decimal feito com Flip-Flop J K

QD	QC	QB	QA	DEC
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

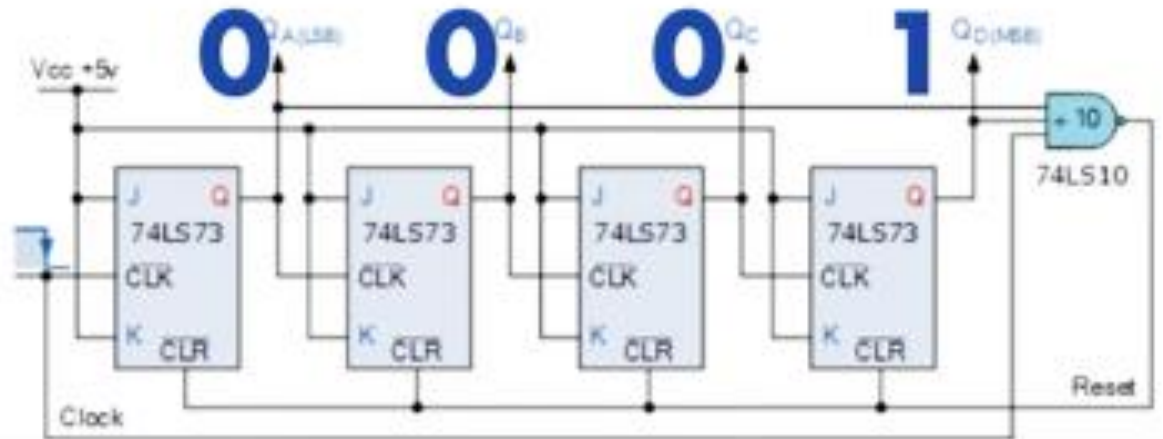


J	K	CLK	Q	Q\
0	0	↓	Q0	Q0\
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	Toggle	

Toggle = Novo estado será o anterior invertido.
Se estava 0 fica 1, se estava 1 fica 0

Contador decimal feito com Flip-Flop J K

QD	QC	QB	QA	DEC
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9

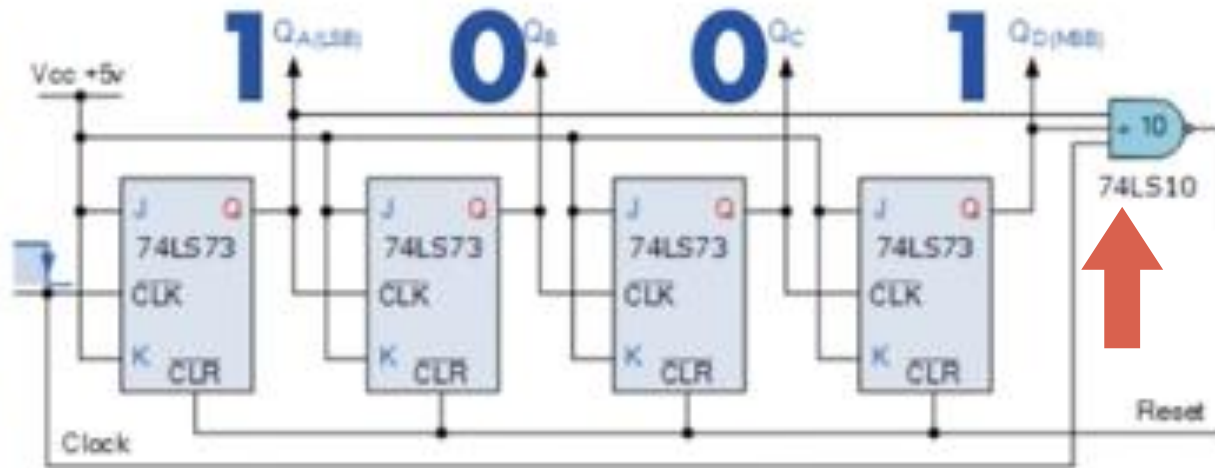


J	K	CLK	Q	Q\
0	0	↓	Q0	Q0\
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	Toggle	

Toggle = Novo estado será o anterior invertido.
Se estava 0 fica 1, se estava 1 fica 0

Contador decimal feito com Flip-Flop J K

QD	QC	QB	QA	DEC
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9



J	K	CLK	Q	Q\
0	0	↓	Q0	Q0\
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	Toggle	

Toggle = Novo estado será o anterior invertido.
Se estava 0 fica 1, se estava 1 fica 0

Contadores Assíncronos

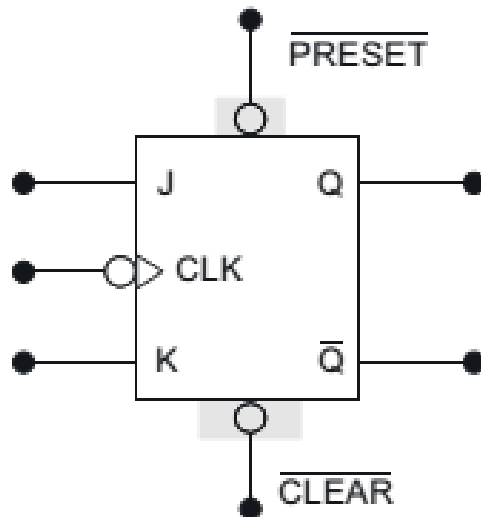
Aplicações:

Um contador é necessário para contar o número de itens que passam por uma esteira em uma fábrica. Uma fotocélula e um feixe de luz são usados para **gerar um pulso** único cada vez que um item passa pelo local. O contador deve permitir a contagem de pelo menos 1000 objetos. Quantos flip-flops são necessários?

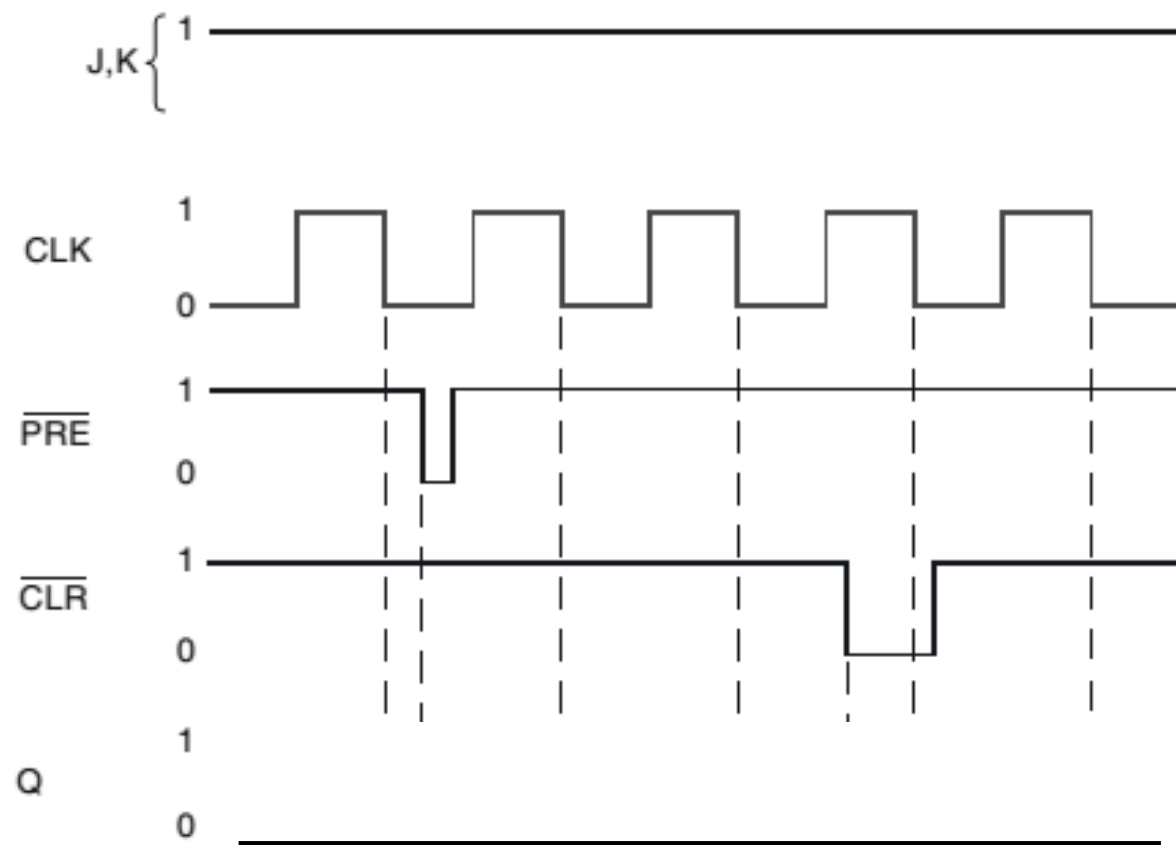
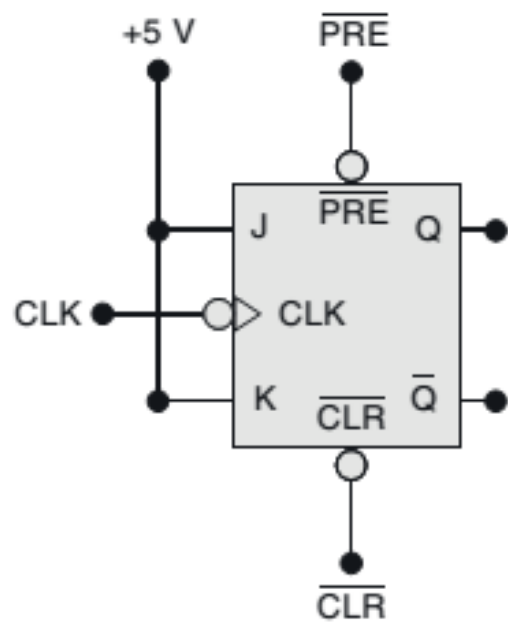
$$2^N \geq 1000$$
$$2^{10} = 1024 \geq 1000$$

ENTRADAS ASSÍNCRONAS

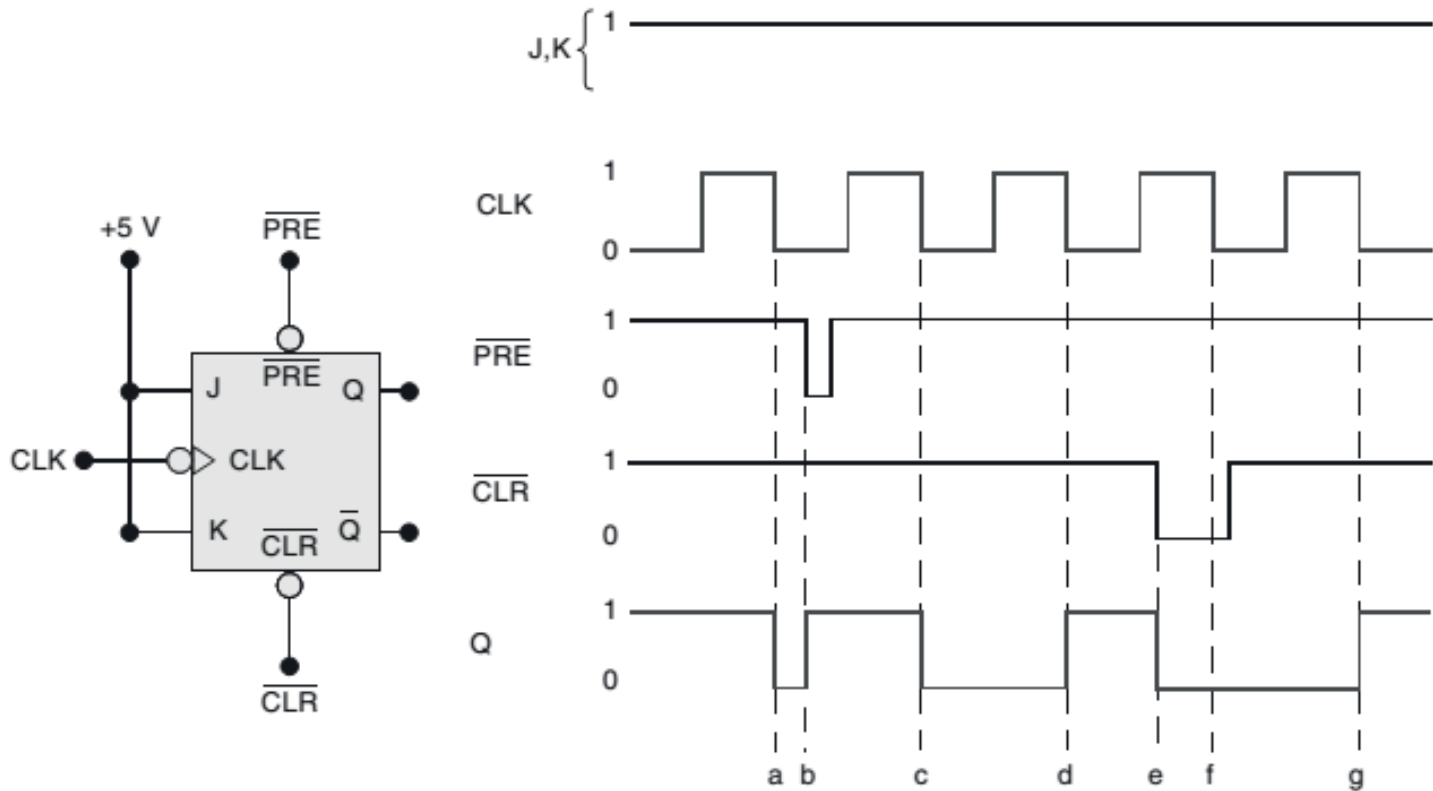
- FFs com clock, podem ter uma ou mais **entradas assíncronas** que operam independentemente das entradas síncronas e do clock.
- Estas entradas podem ser usadas para colocar o FF no estado 1 ou 0 *em qualquer instante, independentemente das condições das outras entradas.*



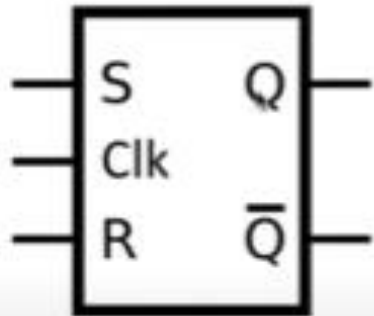
J	K	Clk	$\overline{\text{PRE}}$	$\overline{\text{CLR}}$	Q
0	0	\downarrow	1	1	Q (não muda)
0	1	\downarrow	1	1	0 (reset síncrono)
1	0	\downarrow	1	1	1 (set síncrono)
1	1	\downarrow	1	1	\overline{Q} (toggle síncrono ou comutação síncrona)
x	x	x	1	1	Q (não muda)
x	x	x	1	0	0 (clear assíncrono)
x	x	x	0	1	1 (preset assíncrono)
x	x	x	0	0	(Inválido)



ENTRADAS ASSÍNCRONAS



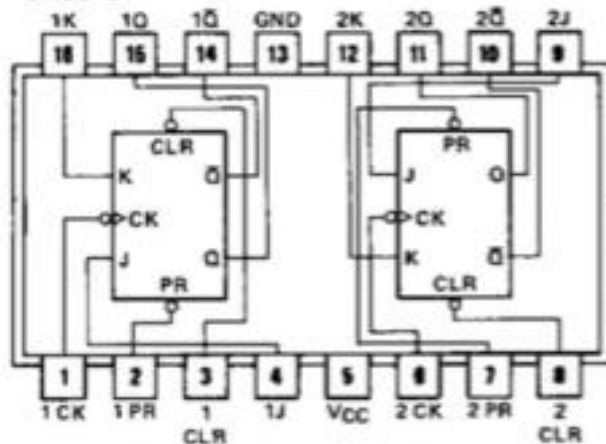
Flip-Flop S R



S	R	Q	Q\
0	0	1 ou 0	0 ou 1
0	1	0	1
1	0	1	0
1	1	Inválido	Inválido

Flip-Flop J K

7476

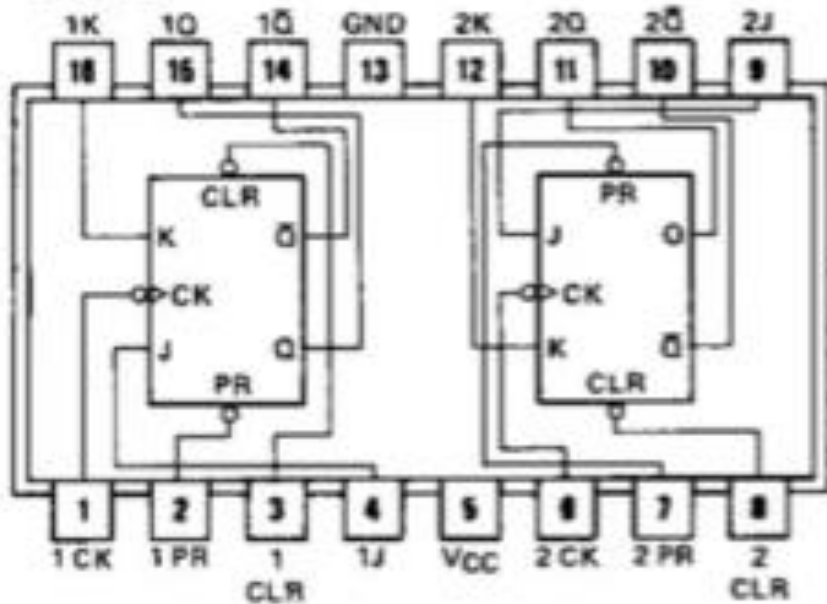


J	K	CLK	Q	Q\
0	0	↓	Q0	Q0\
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	Toggle	

Toggle = Novo estado será o anterior invertido.
Se estava 0 fica 1, se estava 1 fica 0

Flip-Flop J K

7476

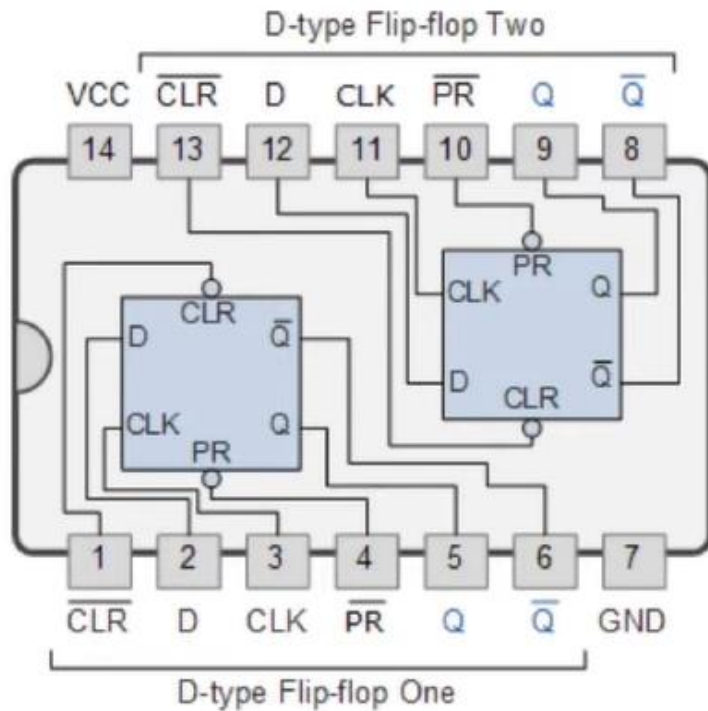


J	K	CLK	Q	Q\
0	0	↓	Q0	Q0\
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	Toggle	

Toggle = Novo estado será o anterior invertido.
Se estava 0 fica 1, se estava 1 fica 0

CLR E PR são prioritários em relação a todos os outros pinos

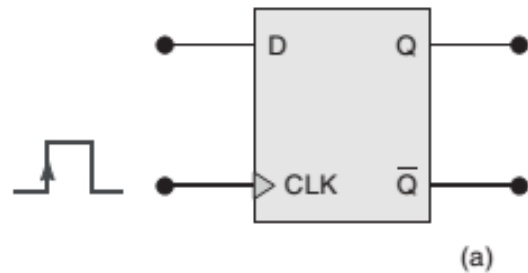
Flip-Flop D



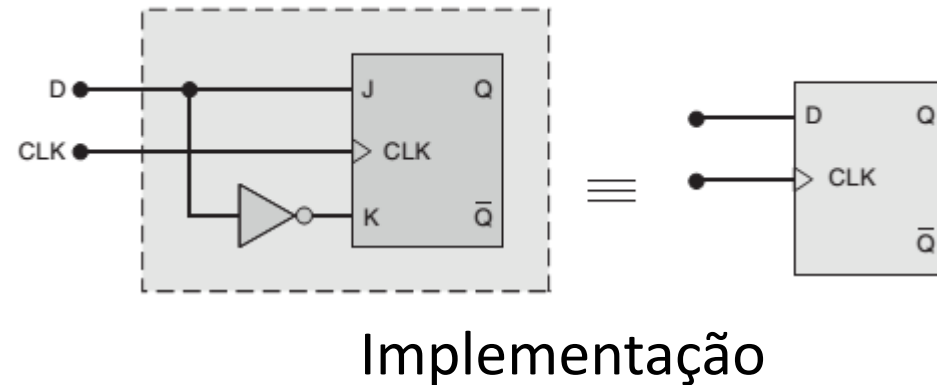
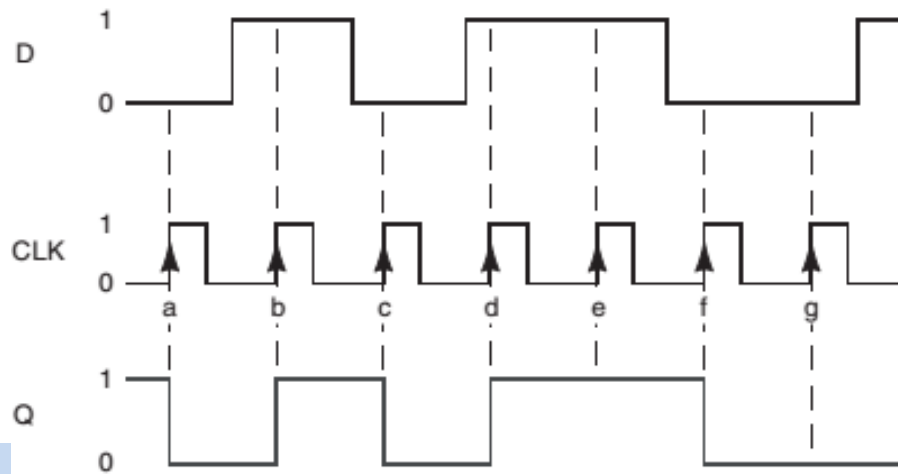
D	CLK	Q	Q\
0	↓	0	1
1	↓	1	0

Flip Flop D com Clock

- A saída Q irá para o mesmo estado lógico presente na entrada D quando ocorrer uma borda de subida em CLK .

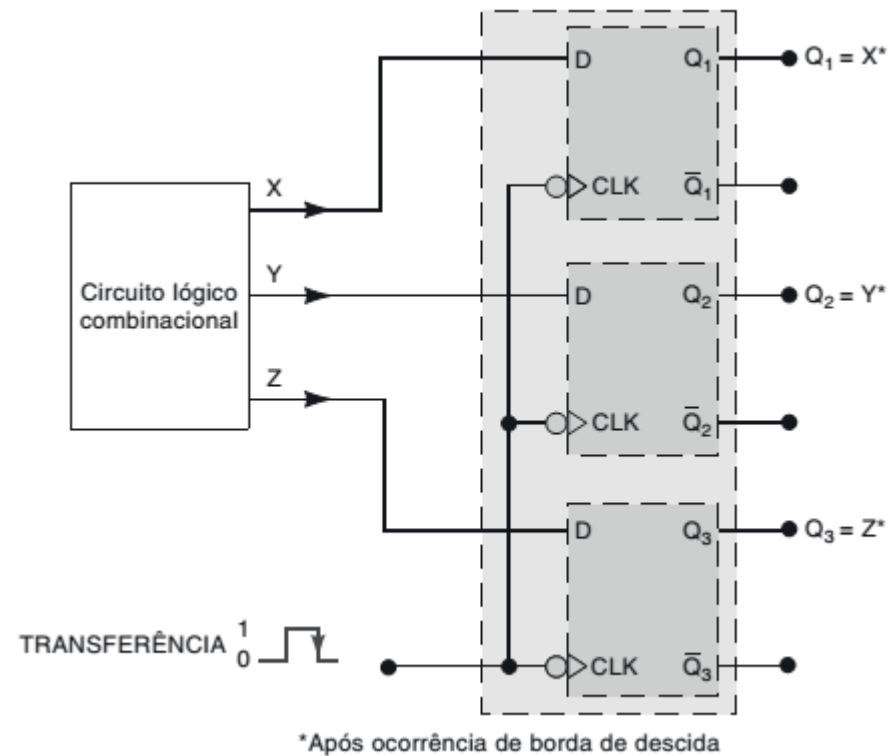


D	CLK	Q
0	↑	0
1	↑	1



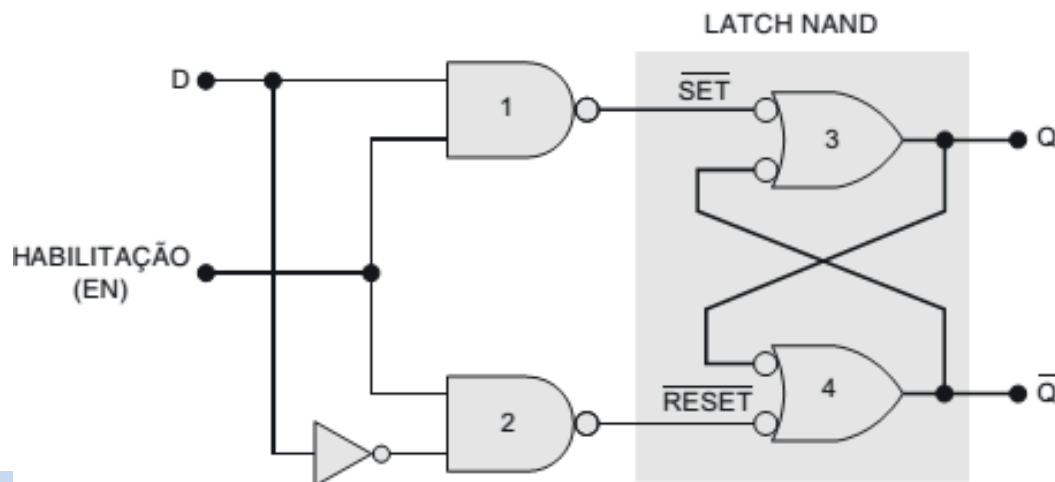
Aplicações do FF Tipo D

- Transferência de Dados em Paralelo
- Usando flip-flops D , os níveis lógicos presentes em X , Y e Z são transferidos para Q_1 , Q_2 e Q_3 , respectivamente, no momento da aplicação do pulso TRANSFERÊNCIA nas entradas CLK comuns.



Latch D Transparente

- O circuito contém um latch NAND e um direcionador (denominada entrada de *habilitação*) de pulsos formado pelas portas NAND no 1 e no 2, **porém não possui o circuito detector de borda.**
- A entrada comum das portas que implementam o circuito direcionador é denominada entrada de *habilitação*.



Entradas		Saída
EN	D	Q
0	X	Q_0 (não muda)
1	0	0
1	1	1

'X' indica irrelevante.
 Q_0 é o estado imediatamente anterior a EN para o nível BAIXO.

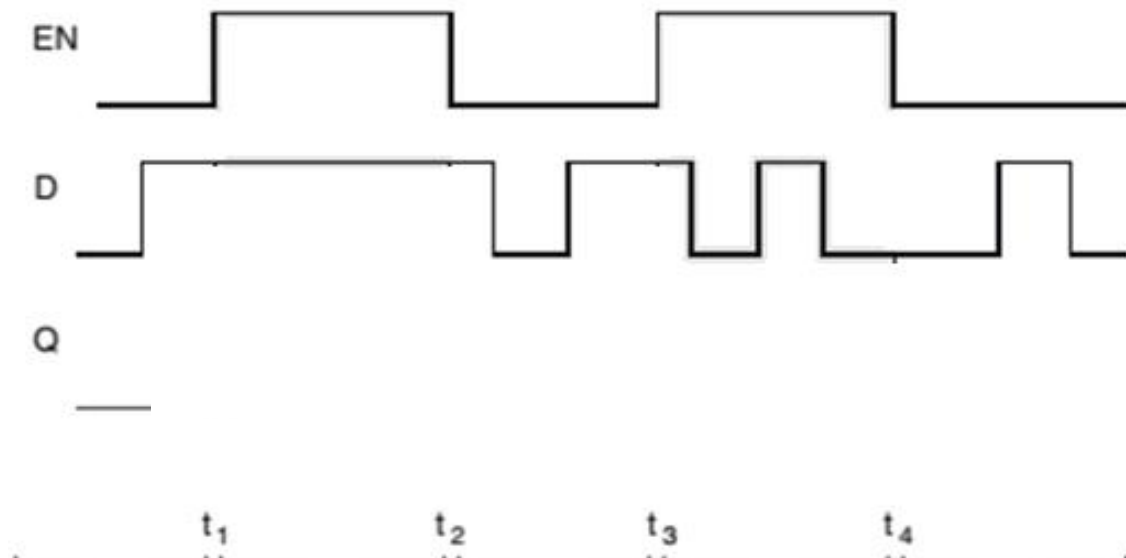
Exemplo

- Determine a forma de onda da saída Q para um latch D com as formas de onda das entradas EN e D mostradas na Figura. Considere inicialmente $Q = 0$.

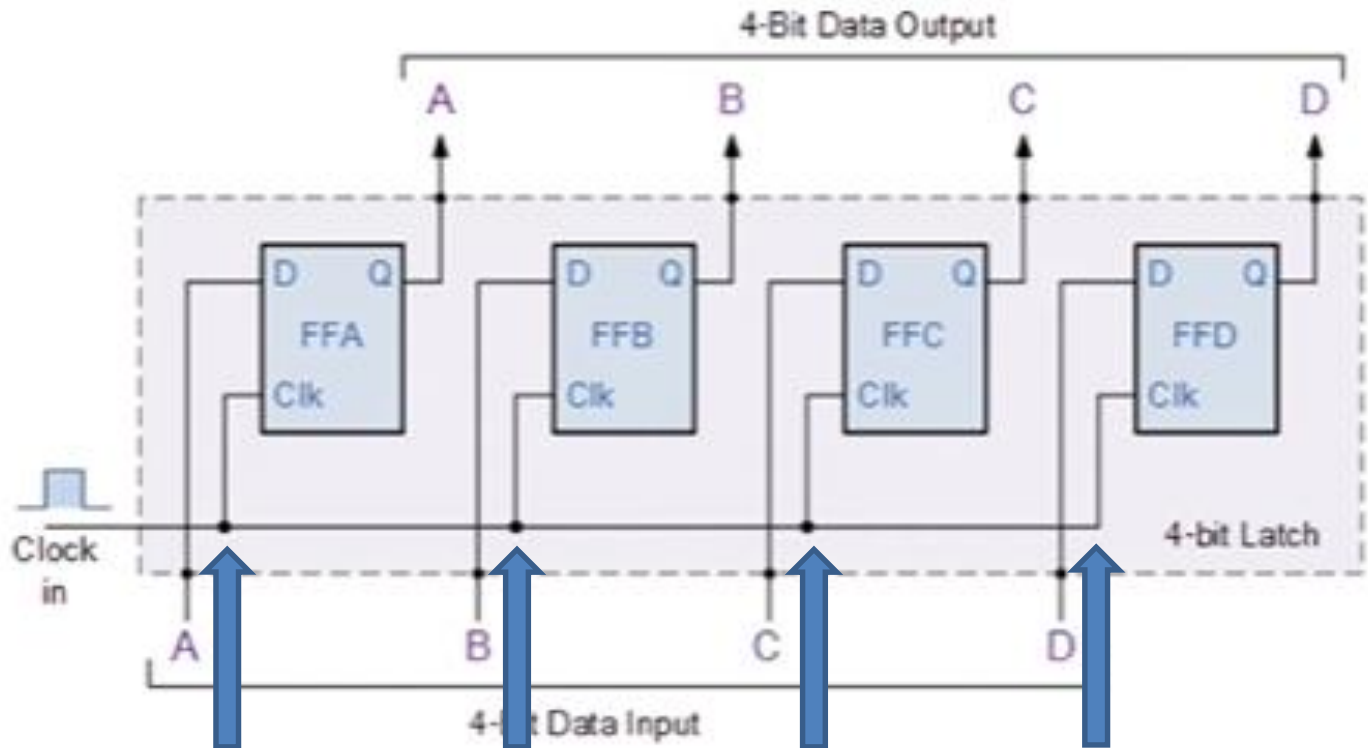
Entradas		Saída
EN	D	Q
0	X	Q_0 (não muda)
1	0	0
1	1	1

'X' indica irrelevante.

Q_0 é o estado imediatamente anterior a EN para o nível BAIXO.



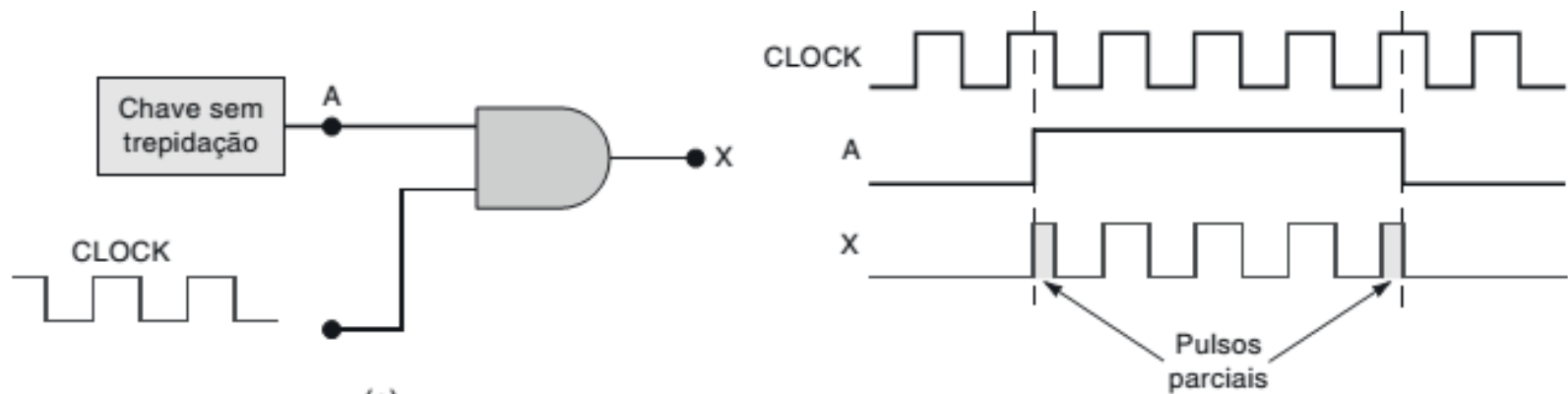
Flip-Flop D – Buffer Paralelo



Sincronização de Flip-Flops

Pulsos parciais

A entrada *A* é usada para controlar a passagem de um sinal de clock por uma porta AND, de modo que os pulsos de clock apareçam na saída *X* apenas quando a entrada *A* estiver em nível ALTO.

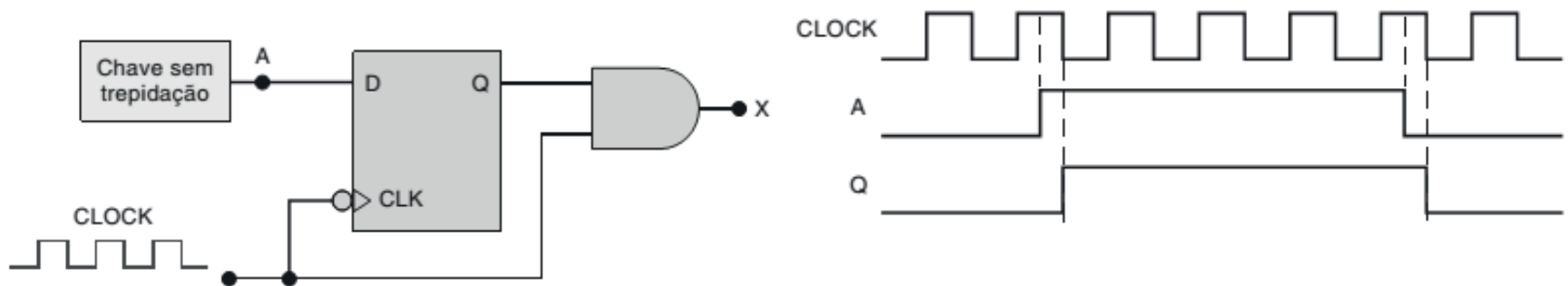


Sincronização de Flip-Flops

Pulsos parciais

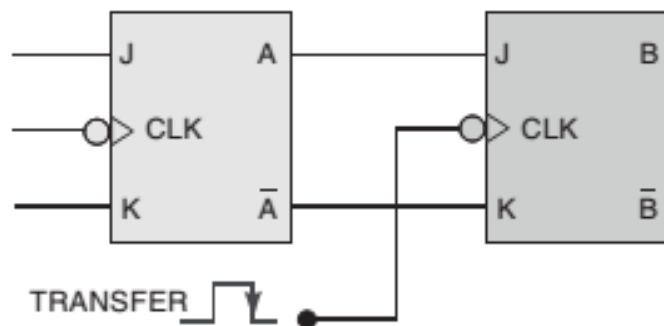
- Solução:

- O sinal no ponto A está conectado à entrada D do flip-flop Q, o qual é disparado pela borda de descida do sinal de clock.
- Quando o ponto A for para o nível ALTO, a saída Q não irá para o nível ALTO até a próxima borda de descida do clock no instante t_1 .
- Esse nível ALTO na saída Q habilita a porta AND a dar passagem ao subsequente pulso *completo* de clock para a saída X

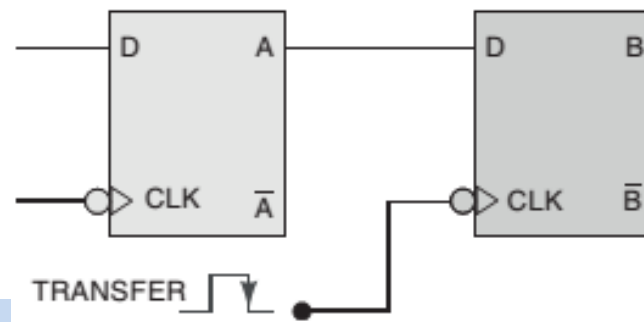


Armazenamento e Transferência de Dados

- O uso mais comum de flip-flops é no armazenamento de dados ou de informações.
- Os FFs que armazenam dados são chamados de **registradores**.
- A **transferência de dados** é definida como a operação de passagem de dados de um FF ou registrador para outro.
- Estas operações podem ser realizadas utilizando o sinal do clock (transferência síncrona) ou através de entradas assíncronas (transferência assíncrona ou transferência por interferência).



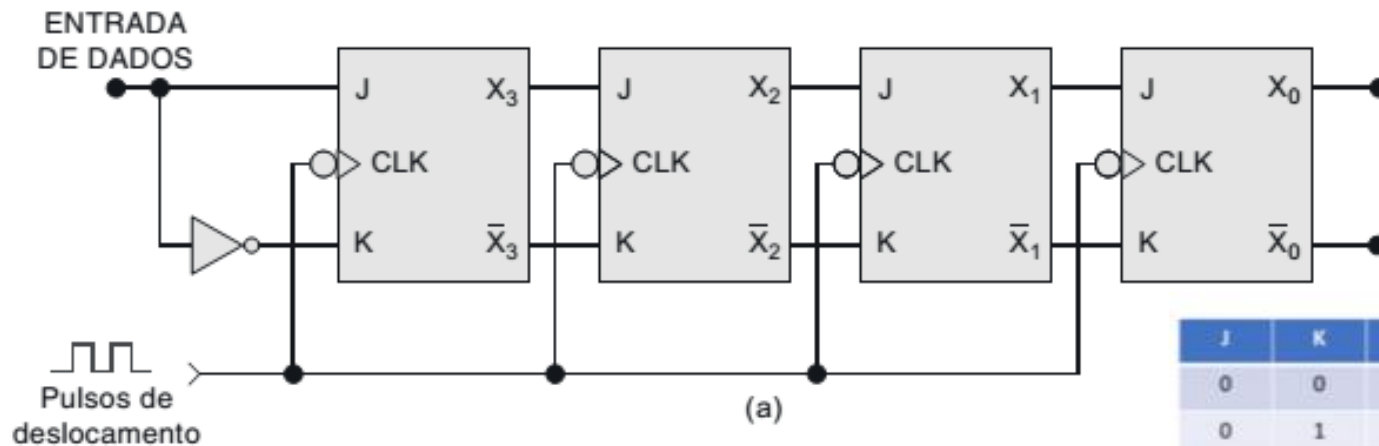
Transferência síncrona



Transferência síncrona

Transferência Serial de Dados: Registradores de Deslocamento

- Um **registrador de deslocamento** é um conjunto de FFs onde os números binários armazenados nos FFs são transmitidos de um FF para o seguinte a cada pulso de clock.



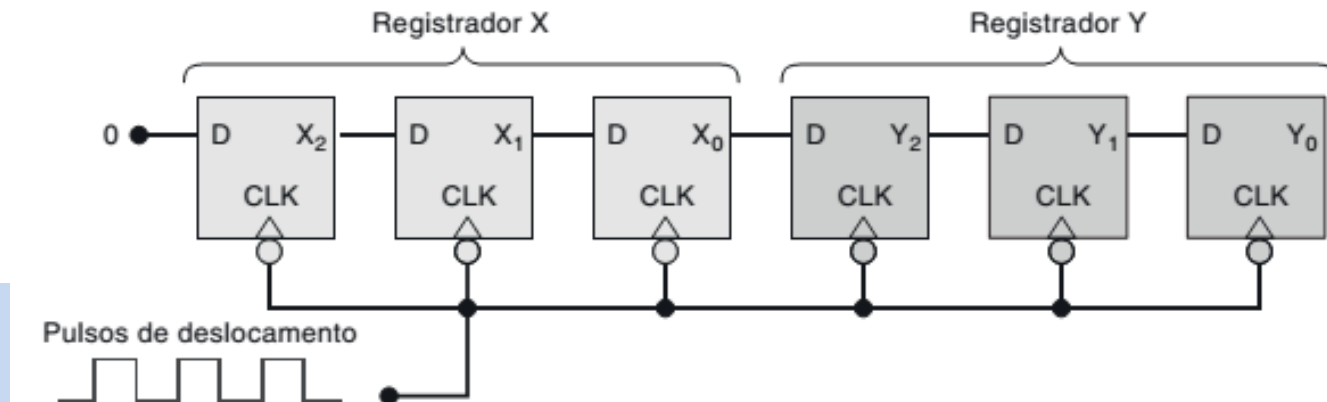
J	K	CLK	Q	Q\
0	0	↓	Q_0	Q_0
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	Toggle	

Toggle = Novo estado será o anterior invertido
Se estava 0 fica 1, se estava 1 fica 0

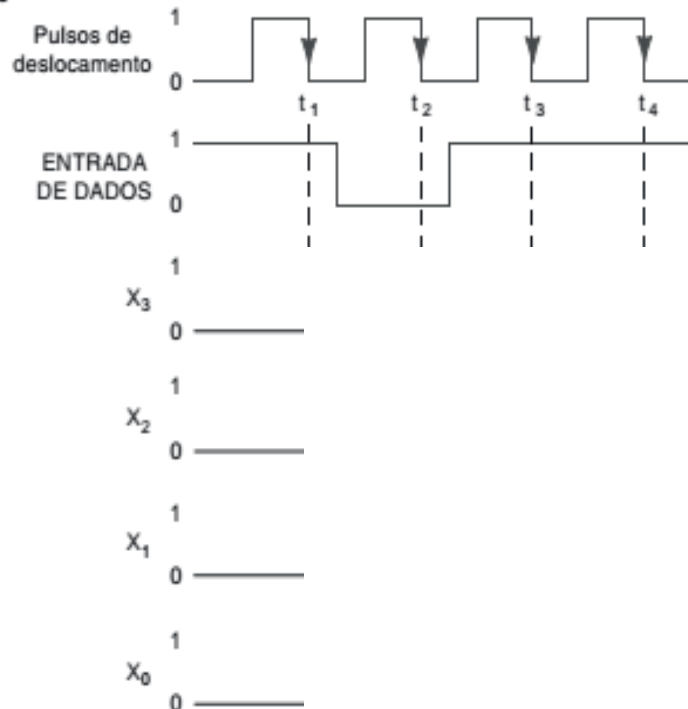
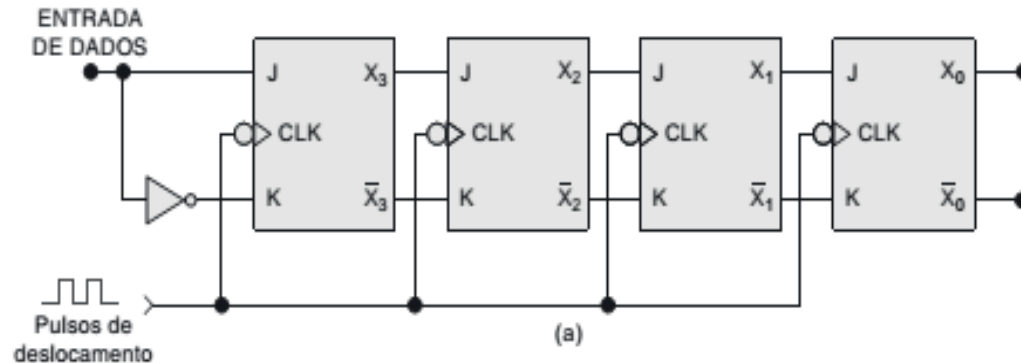
Transferência Serial de Dados: Registradores de Deslocamento

- Transferência serial entre registradores:
- A figura mostra dois registradores de deslocamento de três bits conectados de modo que o conteúdo do registrador X seja transferido de forma serial (deslocada) para o registrador Y.
- Na transferência serial, a transferência completa de N bits de informação requer N pulsos de clock.
- Quando os pulsos de deslocamento são aplicados, a transferência da informação acontece da seguinte maneira:

$X_2 \rightarrow X_1 \rightarrow X_0 \rightarrow Y_2 \rightarrow Y_1 \rightarrow Y_0$.



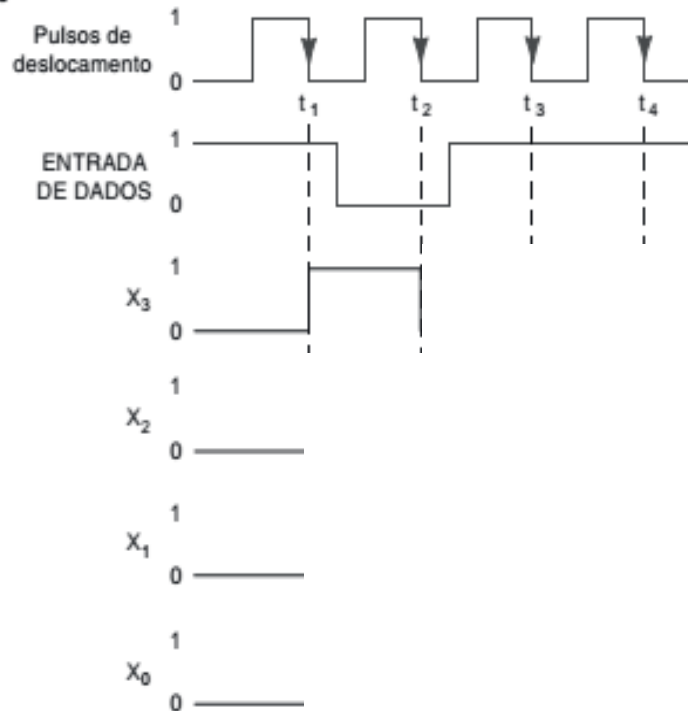
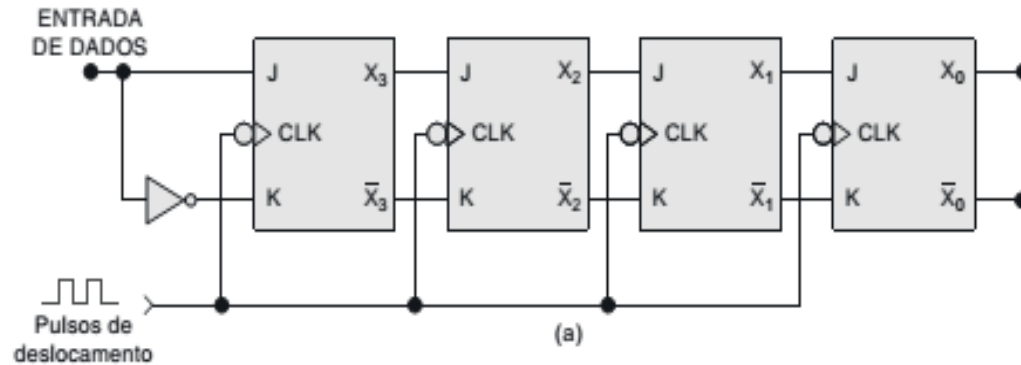
Transferência Serial de Dados: Registradores de Deslocamento



J	K	CLK	Q	Q\
0	0	↓	Q0	Q0\
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	Toggle	

Toggle = Novo estado será o anterior invertido.
Se estava 0 fica 1, se estava 1 fica 0

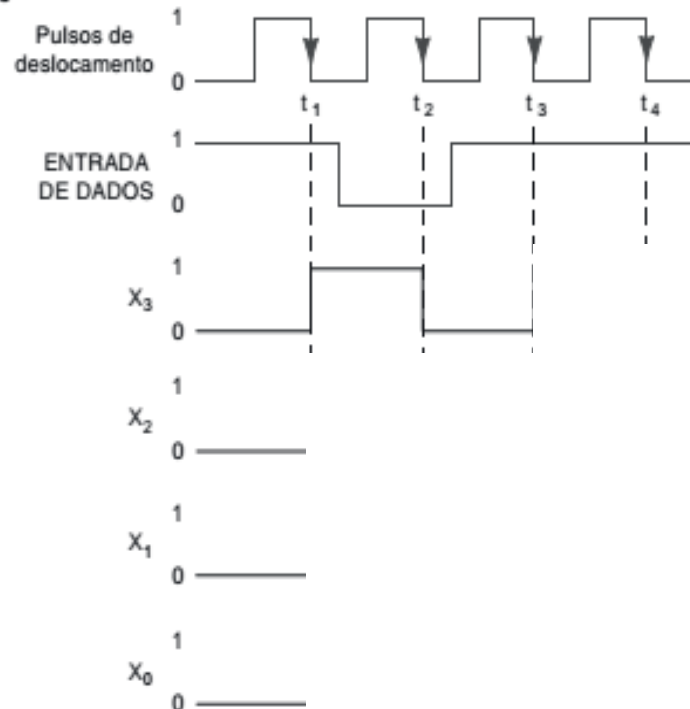
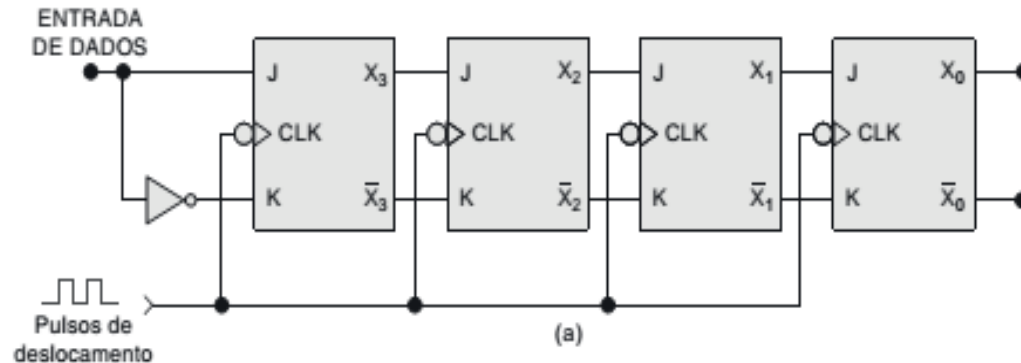
Transferência Serial de Dados: Registradores de Deslocamento



J	K	CLK	Q	Q\
0	0	↓	Q0	Q0\
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	Toggle	

Toggle = Novo estado será o anterior invertido.
Se estava 0 fica 1, se estava 1 fica 0

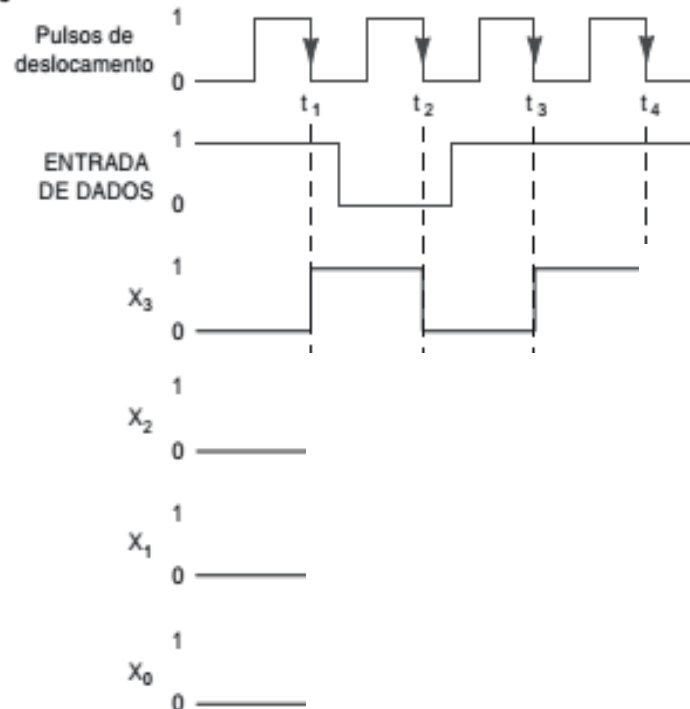
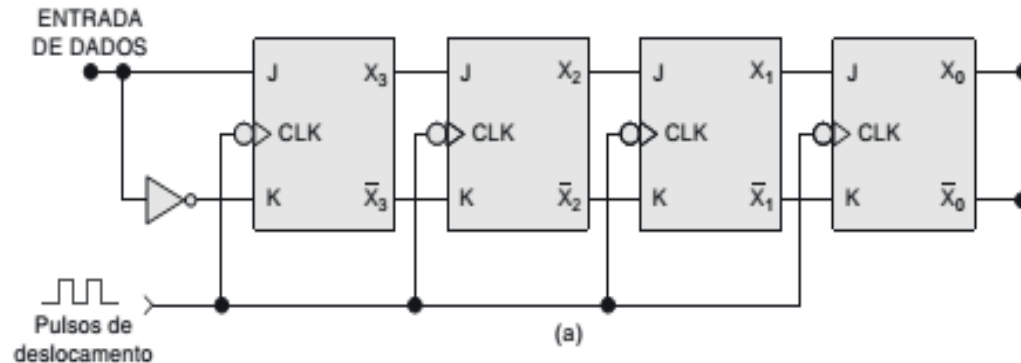
Transferência Serial de Dados: Registradores de Deslocamento



J	K	CLK	Q	Q\
0	0	↓	Q0	Q0\
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	Toggle	

Toggle = Novo estado será o anterior invertido.
Se estava 0 fica 1, se estava 1 fica 0

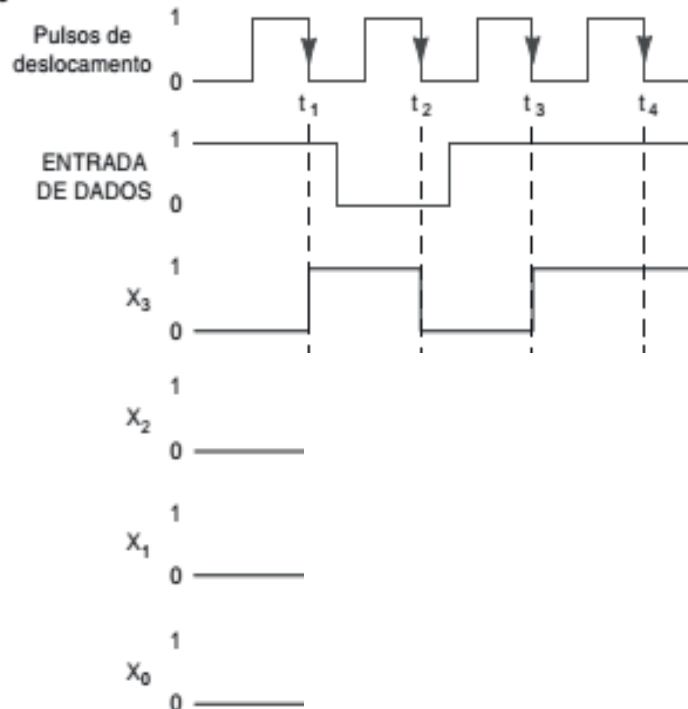
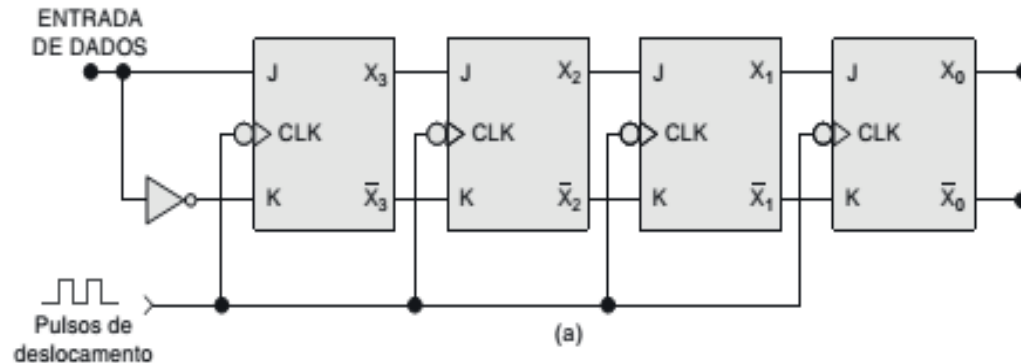
Transferência Serial de Dados: Registradores de Deslocamento



J	K	CLK	Q	Q\
0	0	↓	Q_0	$Q_0\backslash$
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	Toggle	

Toggle = Novo estado será o anterior invertido.
Se estava 0 fica 1, se estava 1 fica 0

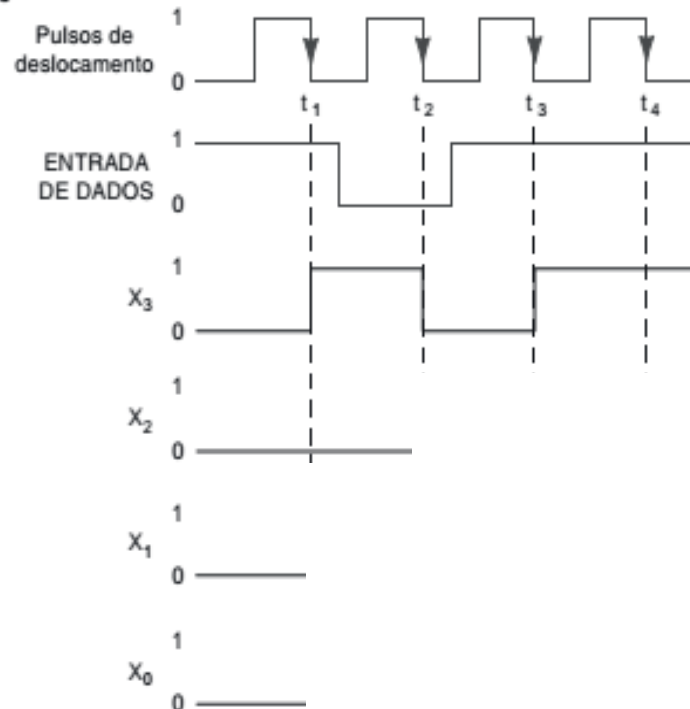
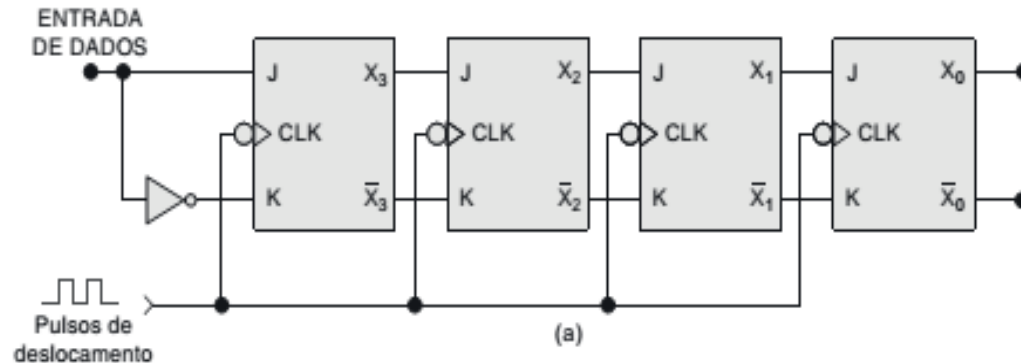
Transferência Serial de Dados: Registradores de Deslocamento



J	K	CLK	Q	Q\
0	0	↓	Q0	Q0\
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	Toggle	

Toggle = Novo estado será o anterior invertido.
Se estava 0 fica 1, se estava 1 fica 0

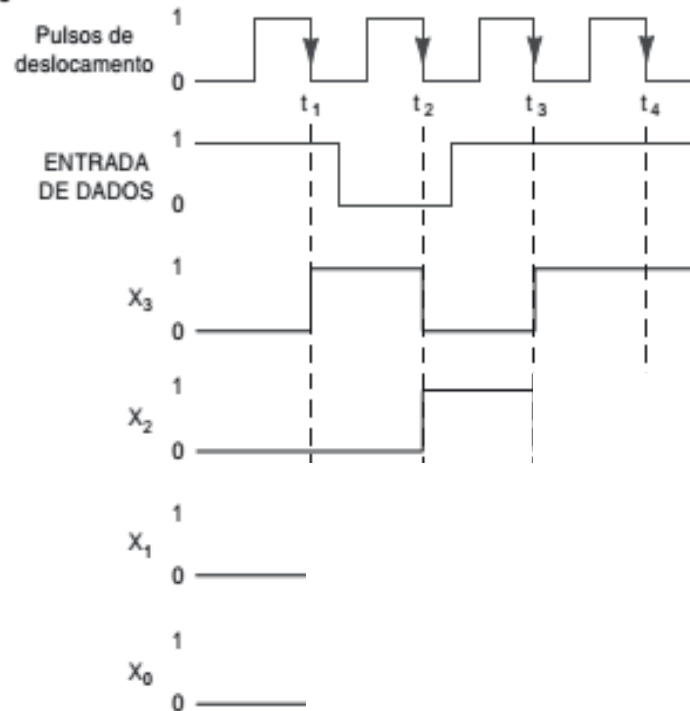
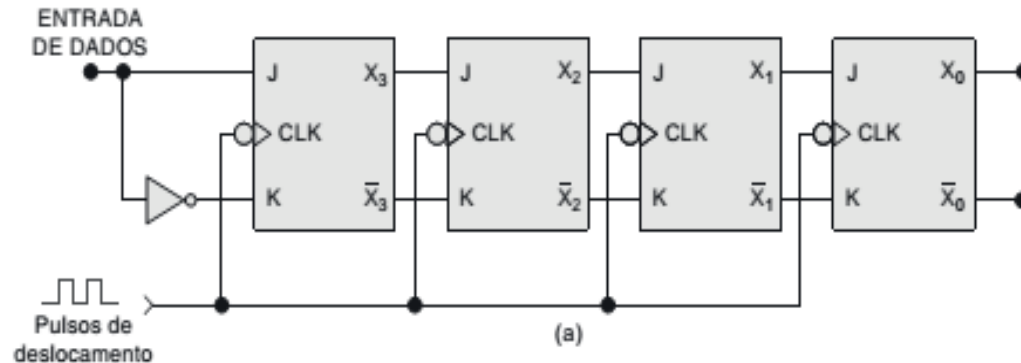
Transferência Serial de Dados: Registradores de Deslocamento



J	K	CLK	Q	Q\
0	0	↓	Q0	Q0\
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	Toggle	

Toggle = Novo estado será o anterior invertido.
Se estava 0 fica 1, se estava 1 fica 0

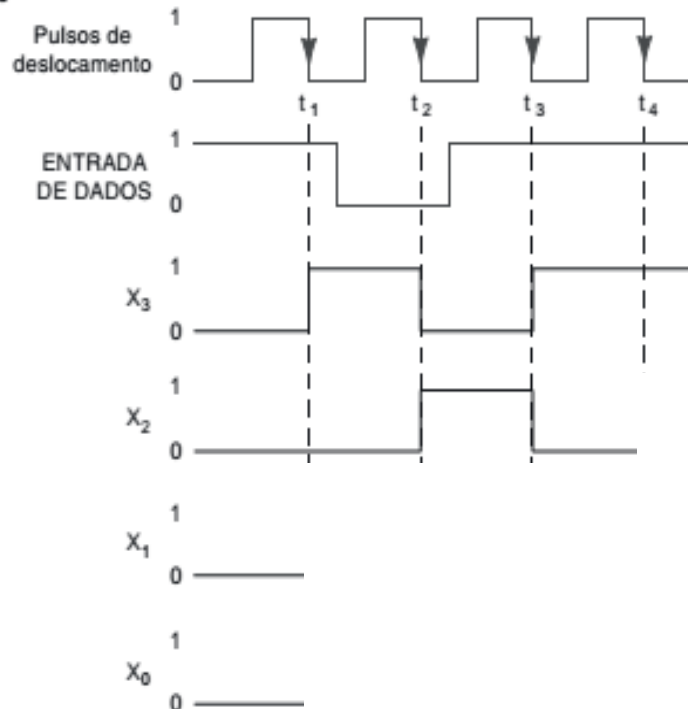
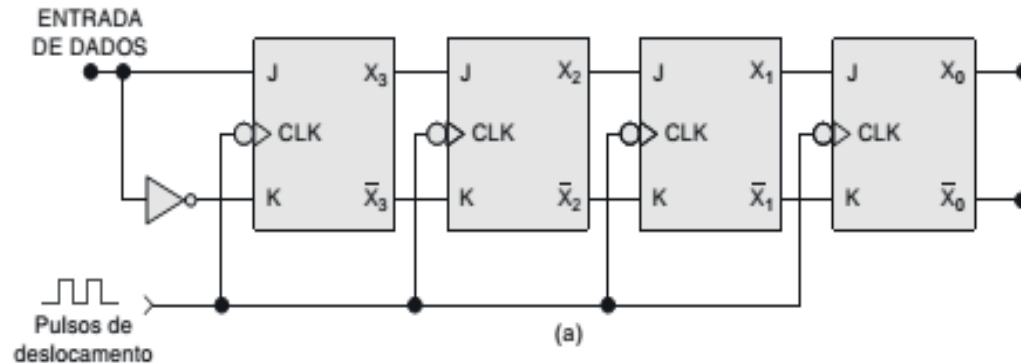
Transferência Serial de Dados: Registradores de Deslocamento



J	K	CLK	Q	Q\
0	0	↓	Q_0	$Q_0\backslash$
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	Toggle	

Toggle = Novo estado será o anterior invertido.
Se estava 0 fica 1, se estava 1 fica 0

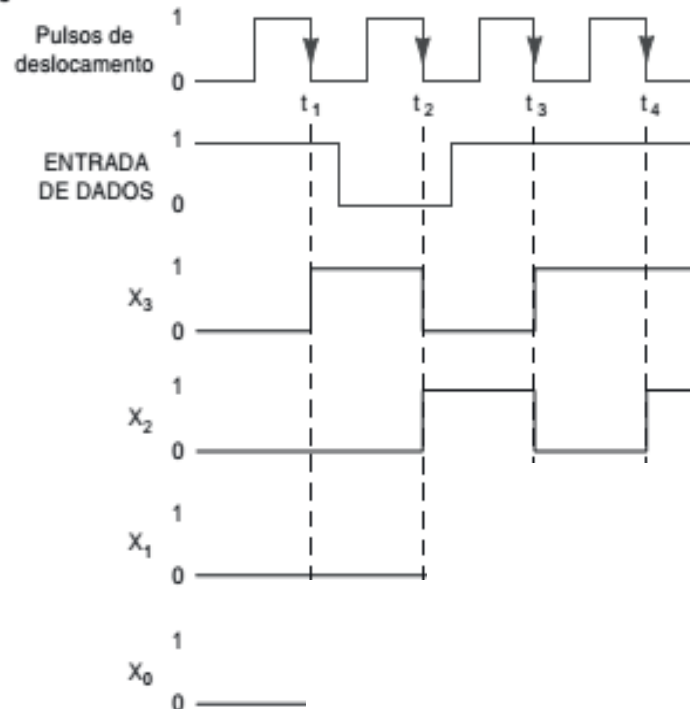
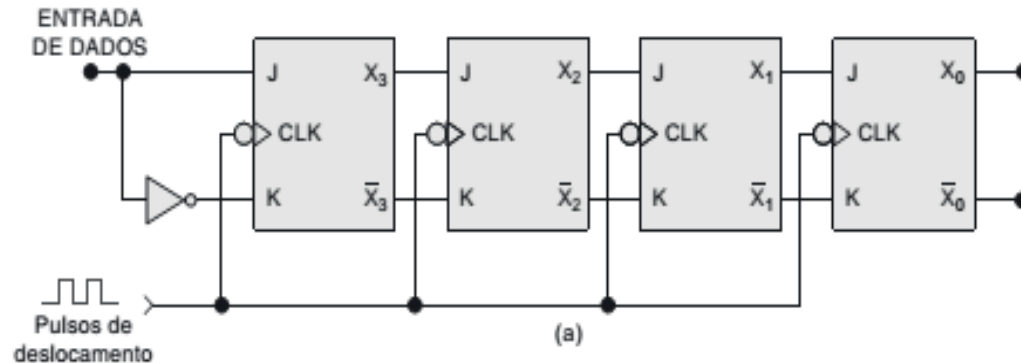
Transferência Serial de Dados: Registradores de Deslocamento



J	K	CLK	Q	Q\
0	0	↓	Q0	Q0\
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	Toggle	

Toggle = Novo estado será o anterior invertido.
Se estava 0 fica 1, se estava 1 fica 0

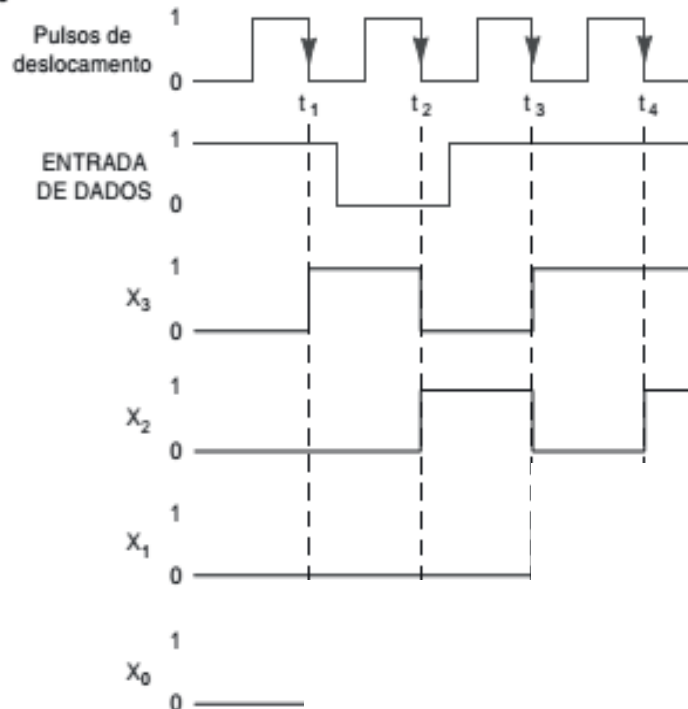
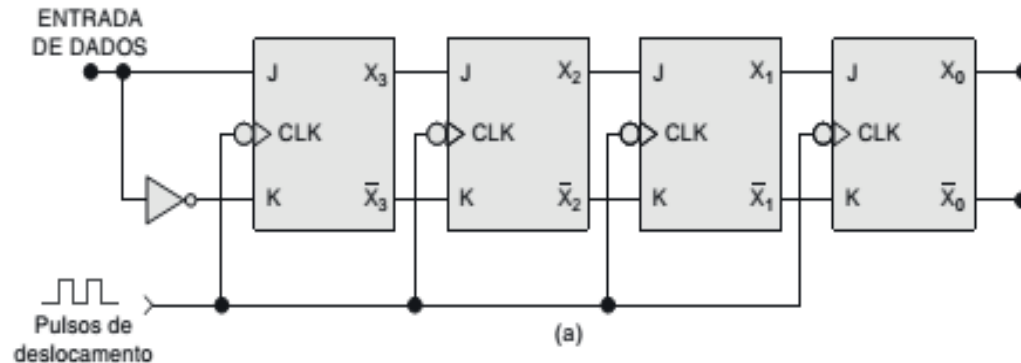
Transferência Serial de Dados: Registradores de Deslocamento



J	K	CLK	Q	Q\
0	0	↓	Q0	Q0\
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	Toggle	

Toggle = Novo estado será o anterior invertido.
Se estava 0 fica 1, se estava 1 fica 0

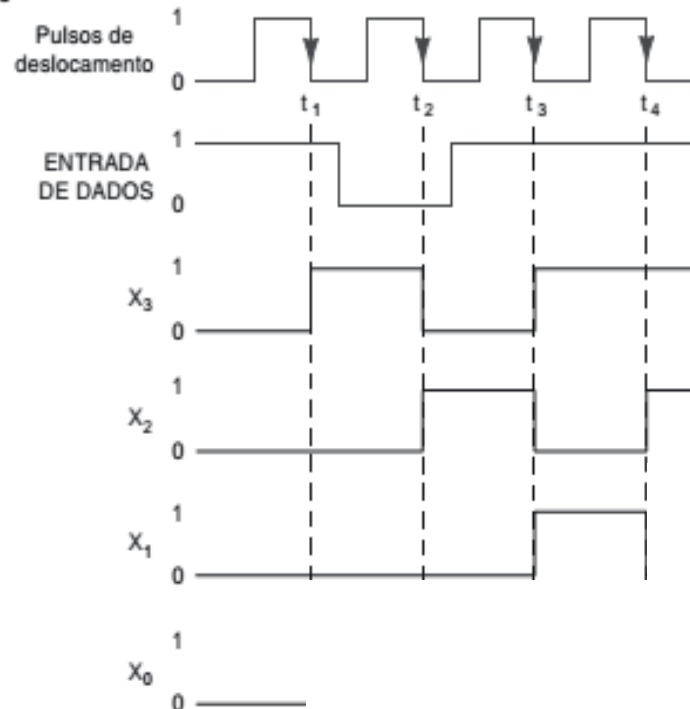
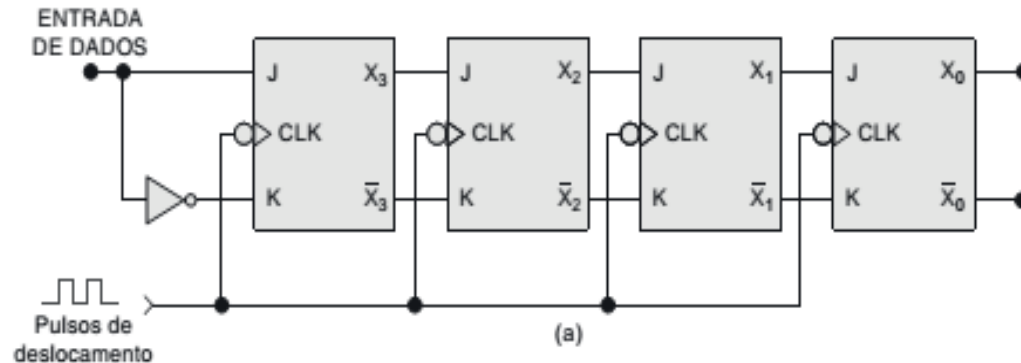
Transferência Serial de Dados: Registradores de Deslocamento



J	K	CLK	Q	Q\
0	0	↓	Q0	Q0\
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	Toggle	

Toggle = Novo estado será o anterior invertido.
Se estava 0 fica 1, se estava 1 fica 0

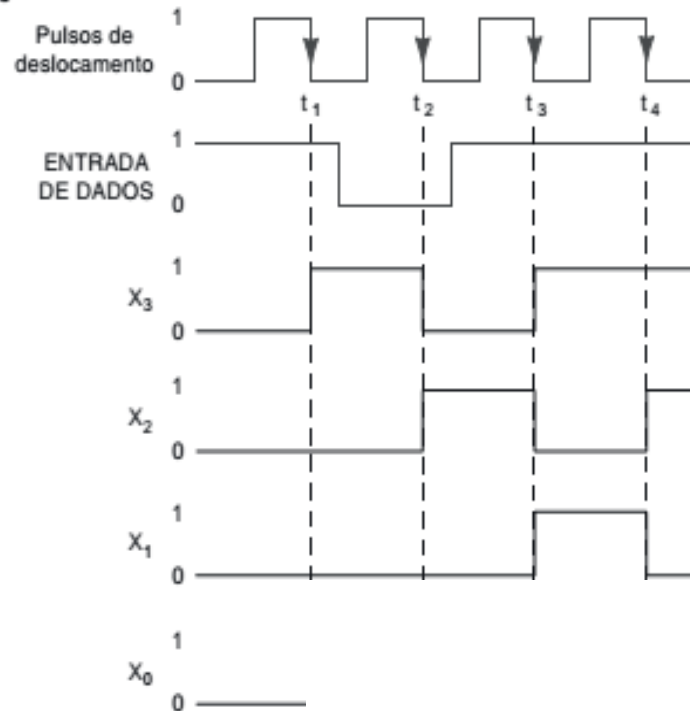
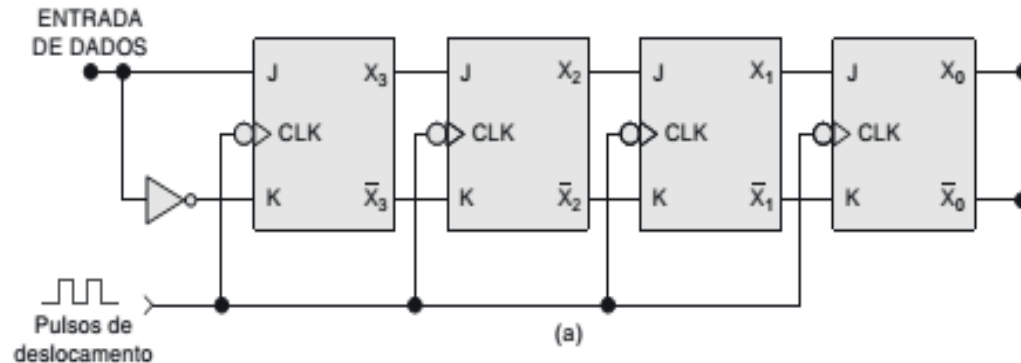
Transferência Serial de Dados: Registradores de Deslocamento



J	K	CLK	Q	Q\
0	0	↓	Q0	Q0\
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	Toggle	

Toggle = Novo estado será o anterior invertido.
Se estava 0 fica 1, se estava 1 fica 0

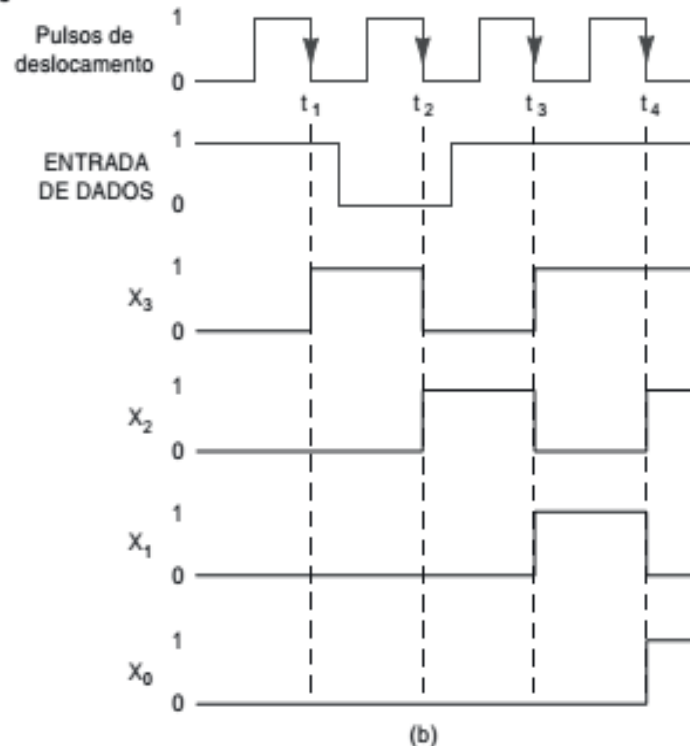
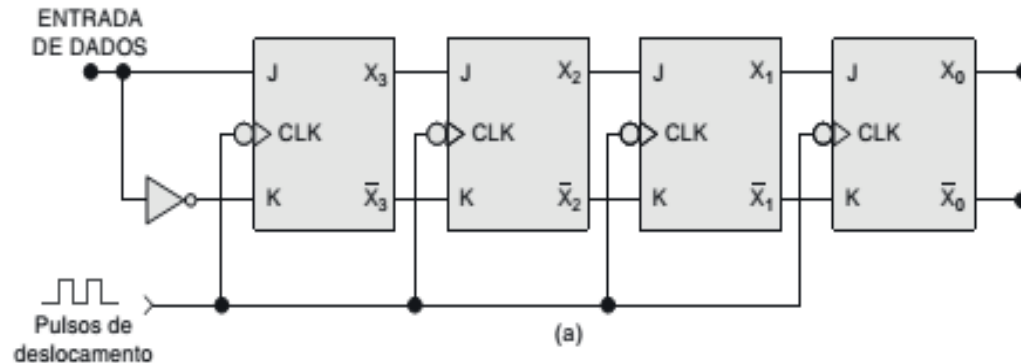
Transferência Serial de Dados: Registradores de Deslocamento



J	K	CLK	Q	Q\
0	0	↓	Q0	Q0\
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	Toggle	

Toggle = Novo estado será o anterior invertido.
Se estava 0 fica 1, se estava 1 fica 0

Transferência Serial de Dados: Registradores de Deslocamento

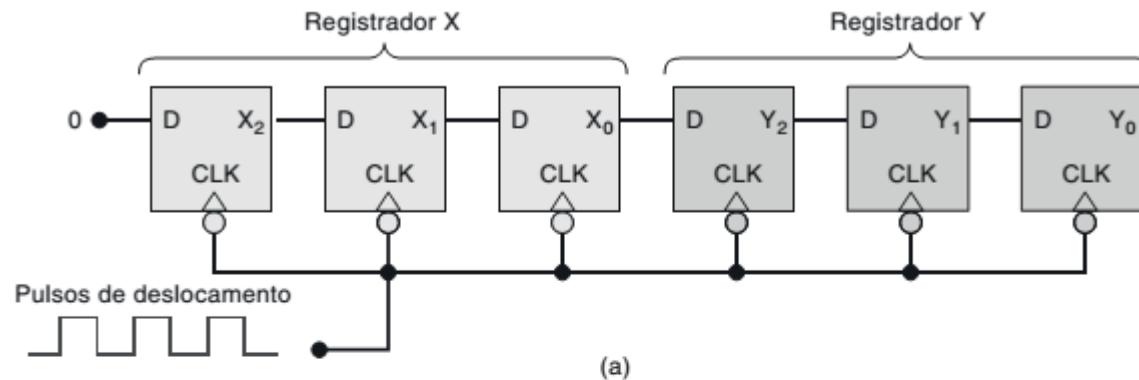


J	K	CLK	Q	Q\
0	0	↓	Q_0	$Q_0\backslash$
0	1	↓	0	1
1	0	↓	1	0
1	1	↓	Toggle	

Toggle = Novo estado será o anterior invertido.
Se estava 0 fica 1, se estava 1 fica 0

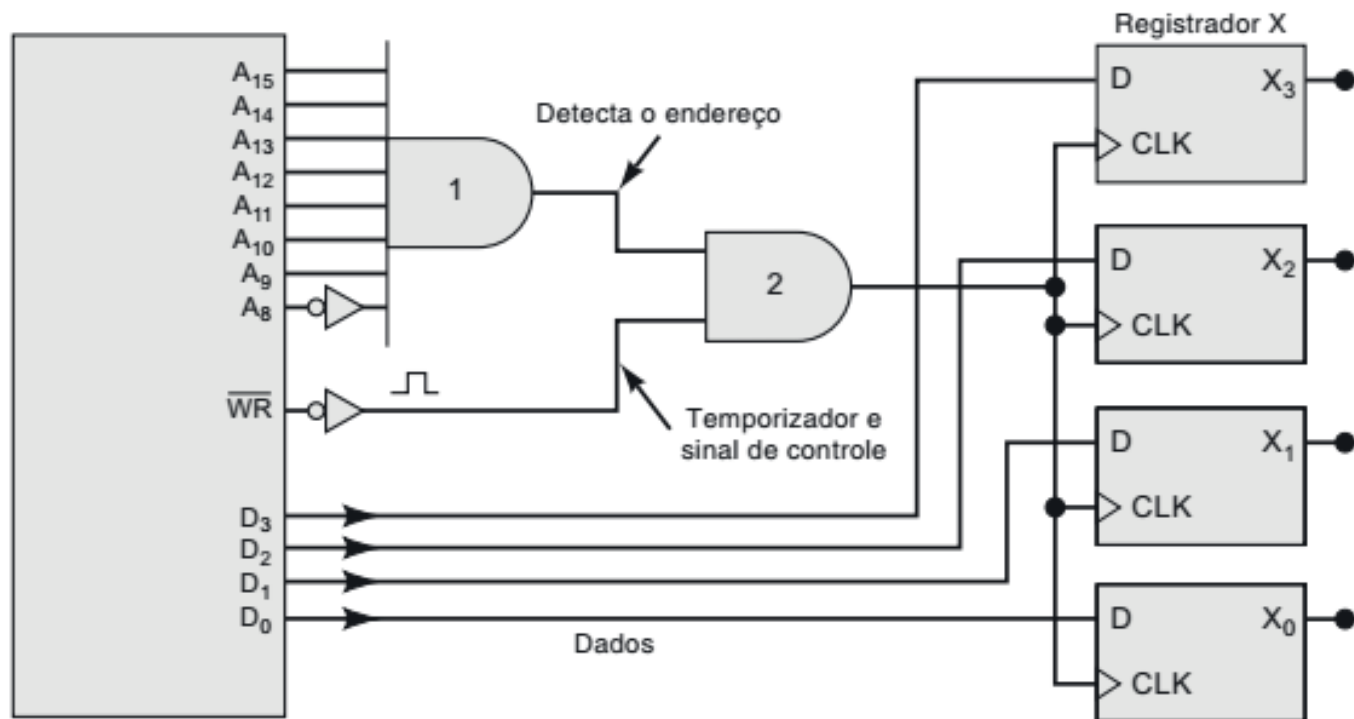
Transferência Serial de Dados: Registradores de Deslocamento

- Transferência serial entre registradores:



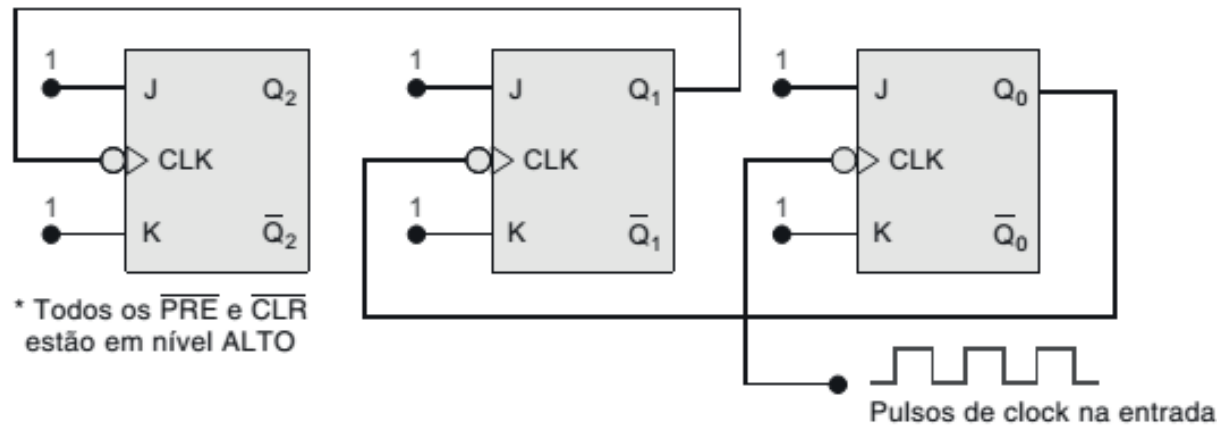
X ₂	X ₁	X ₀	Y ₂	Y ₁	Y ₀
1	0	1	0	0	0
0	1	0	1	0	0
0	0	1	0	1	0
0	0	0	1	0	1

Aplicação em Microcomputador



Divisão de Frequência e Contagem

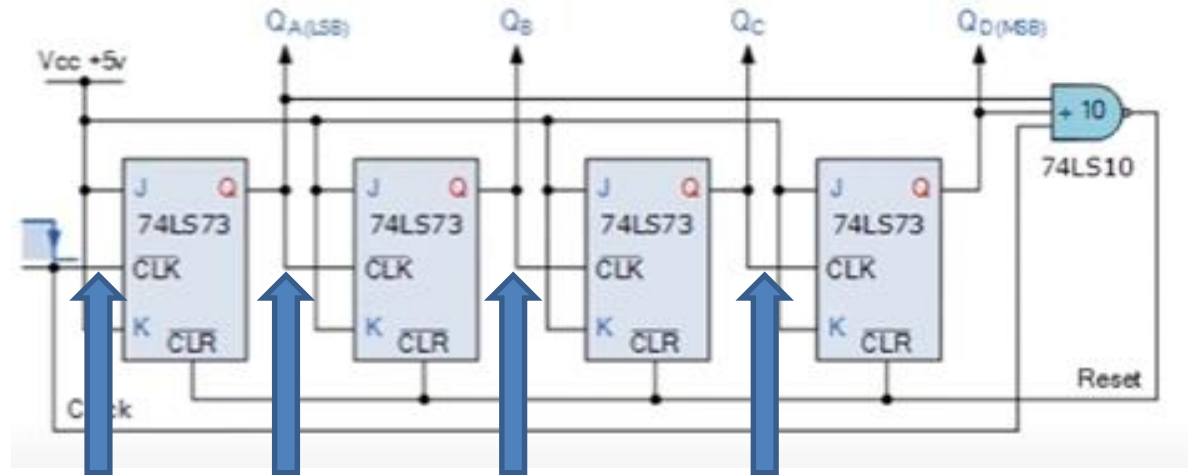
- A divisão de frequência é o processo de diminuir a frequência do clock utilizando FFs.



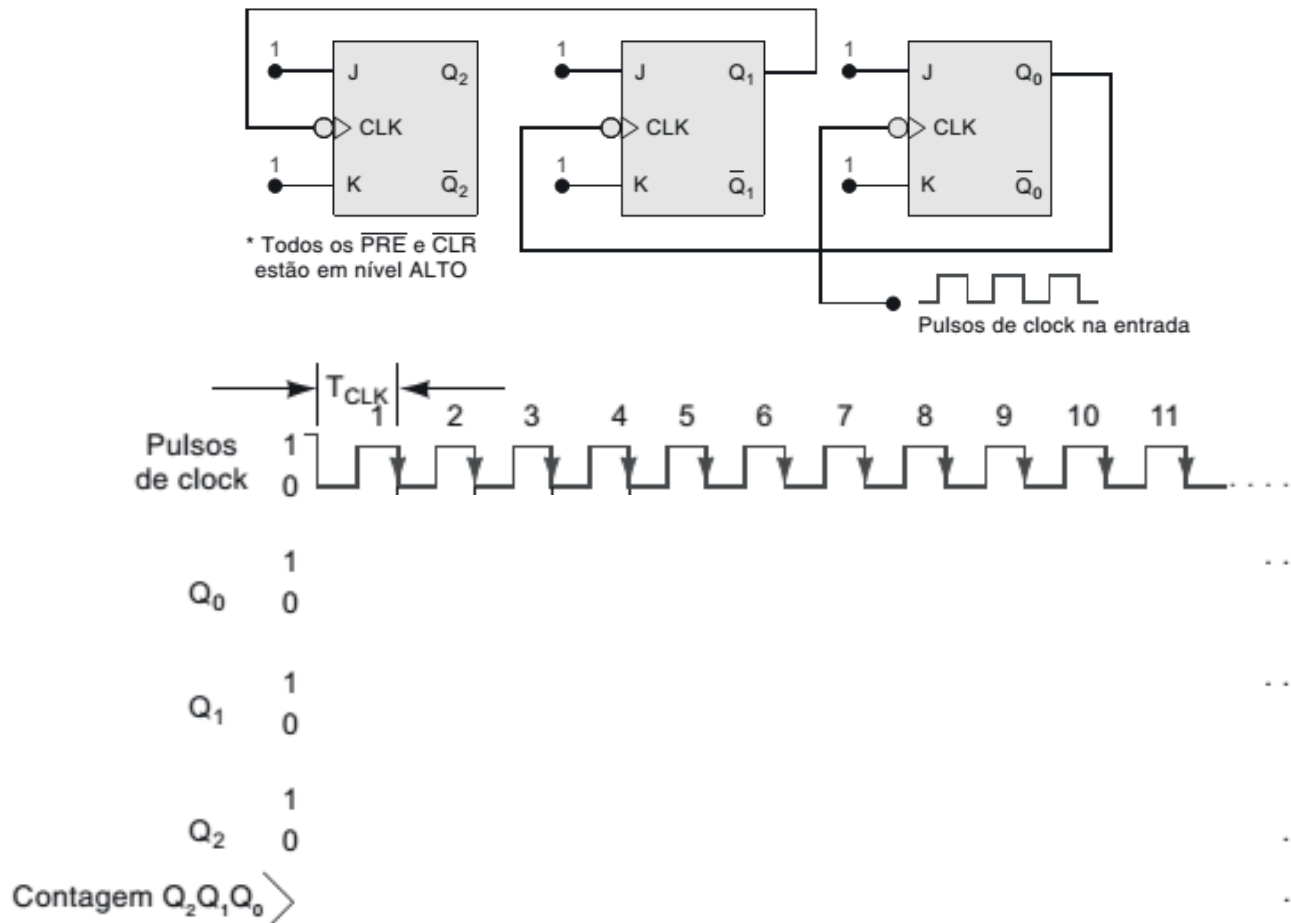
- Cada FF divide a frequência do sinal de sua entrada por 2.

Contador decimal feito com Flip-Flop J K

QD	QC	QB	QA	DEC
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9



DIVISÃO DE FREQUÊNCIA E CONTAGEM



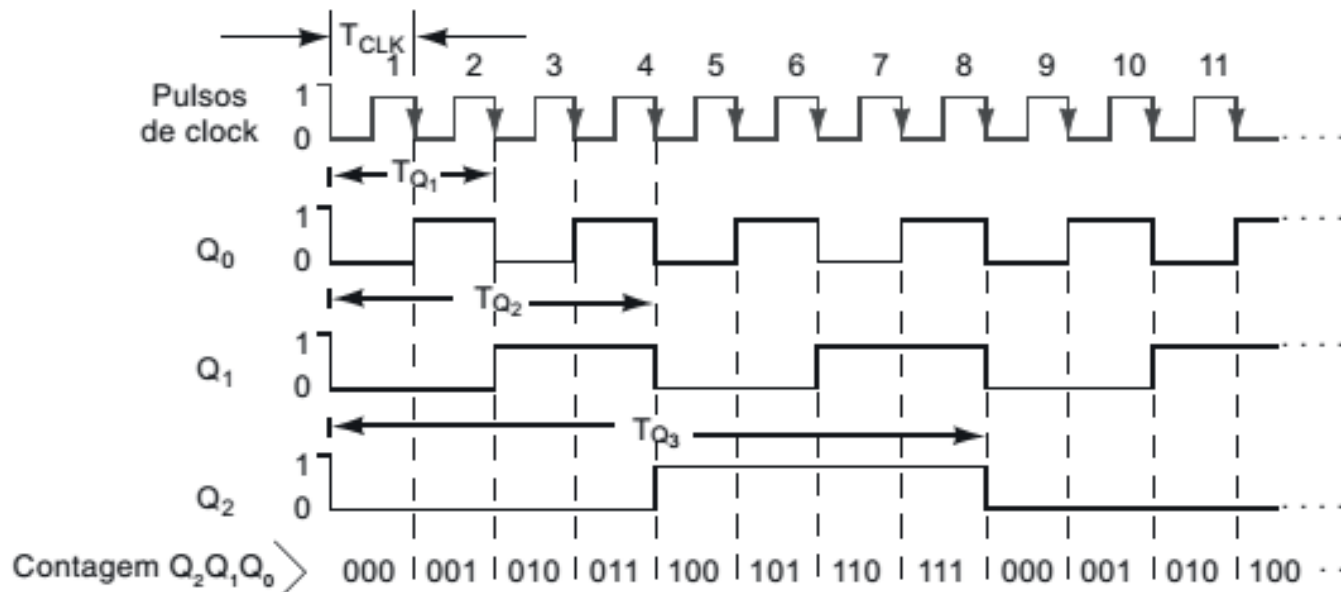
DIVISÃO DE FREQUÊNCIA E CONTAGEM

- Cada FF divide a frequência do sinal de sua entrada por 2.
- Se acrescentarmos um quarto FF a essa cadeia, ele teria frequência igual a $1/2^4 = 1/16$ da frequência de *clock*.
- Usando N flip-flops produziríamos uma frequência de saída no último FF que seria igual a $(1/2^N)$ da frequência de entrada.

DIVISÃO DE FREQUÊNCIA E CONTAGEM

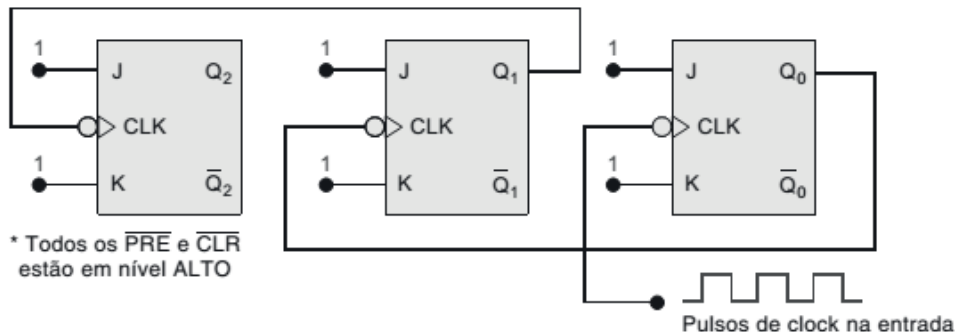
Operação de contagem

- Além de funcionar como divisor de frequência, o circuito também funciona como **contador binário**.

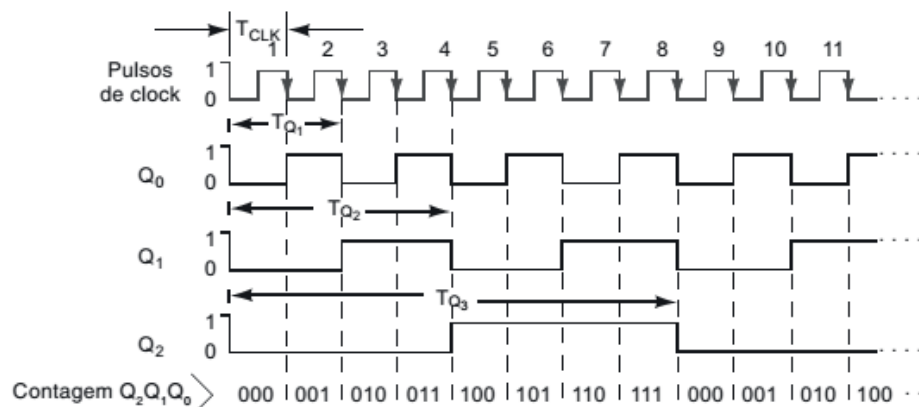


DIVISÃO DE FREQUÊNCIA E CONTAGEM

- Tabela de estados

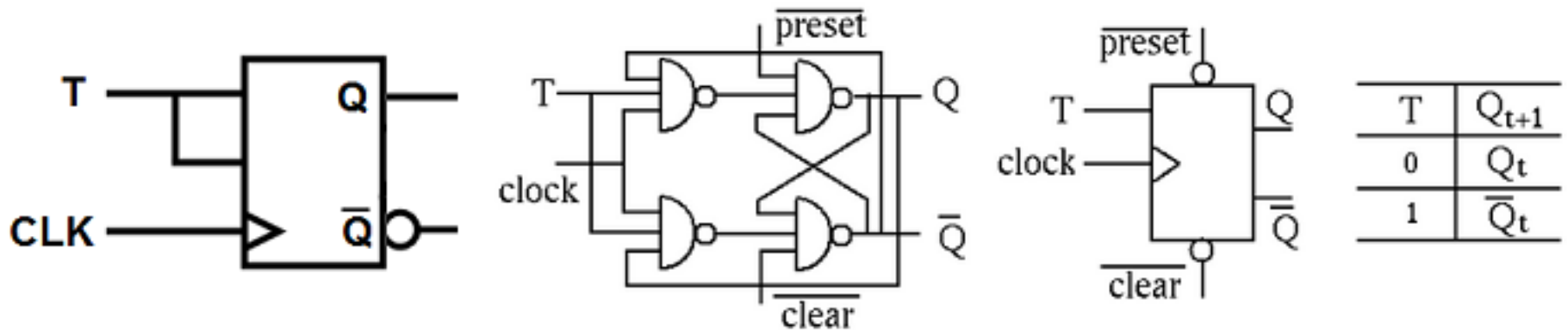


2^2 Q_2	2^1 Q_1	2^0 Q_0	
0	0	0	Antes de aplicar os pulsos de clock
0	0	1	Depois do pulso #1
0	1	0	Depois do pulso #2
0	1	1	Depois do pulso #3
1	0	0	Depois do pulso #4
1	0	1	Depois do pulso #5
1	1	0	Depois do pulso #6
1	1	1	Depois do pulso #7
0	0	0	Depois do pulso #8 retorna para 000
0	0	1	Depois do pulso #9
0	1	0	Depois do pulso #10
0	1	1	Depois do pulso #11
.	.	.	.
.	.	.	.
.	.	.	.

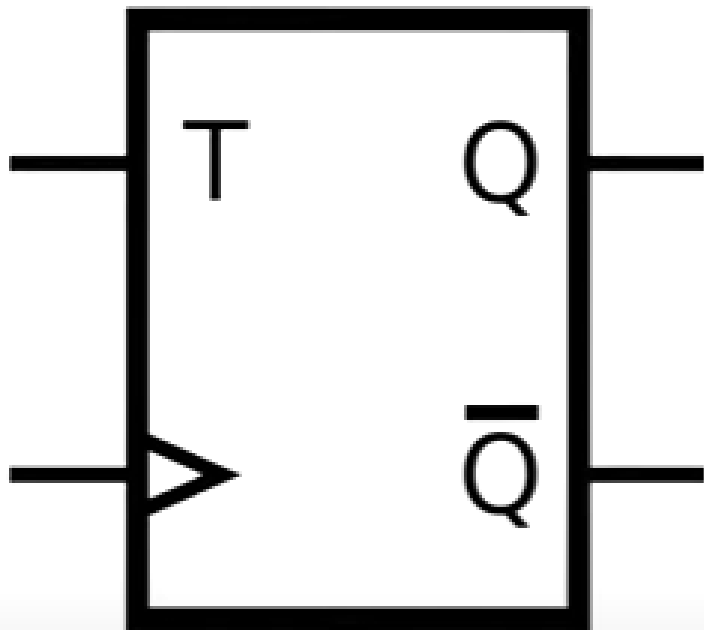


Flip Flop Tipo T

- Um flip-flop T tem uma única entrada (T).
 - Quando $T=0$, o flip-flop está no modo “sem alteração”, similar a um flip-flop J-K com $J=K=0$.
 - Quando $T=1$, o flip-flop está no modo “toggle”, similar a um flip-flop J-K com $J=K=1$.



Flip-Flop T

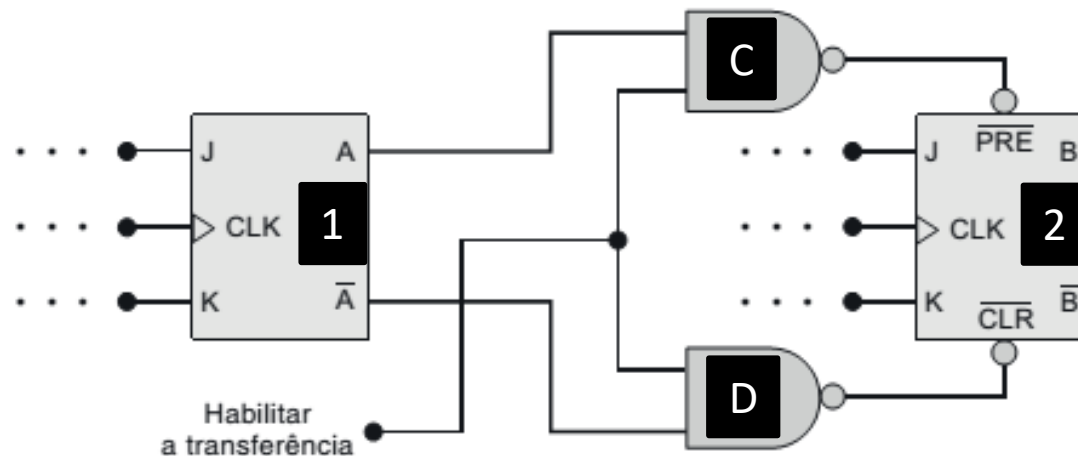


T	CLK	Q	Q\
0	↓	Q0	Q0\
1	↓	Toggle	

“Utiliza” a linha da TV com
entradas 00 e 11 do FF JK

Armazenamento e Transferência de Dados

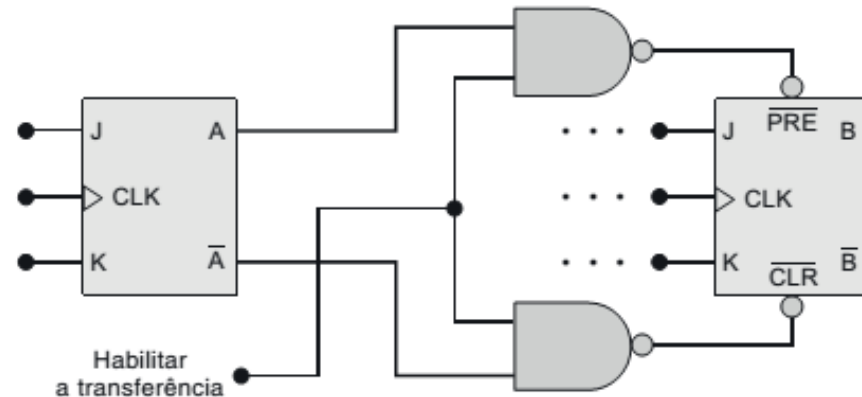
- Analise o seguinte caso:



- A Transferência é síncrona ou assíncrona?
- Descreva seu funcionamento.

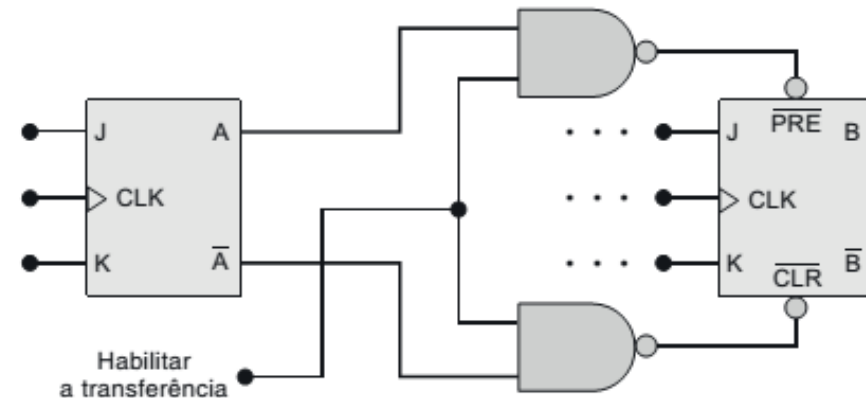
Armazenamento e Transferência de Dados

- Se $\text{TRANSFER ENABLE}=0$; a saída da $\text{NAND}=1$. Não tendo efeito sobre as saídas do FF.



Armazenamento e Transferência de Dados

- Se TRANSFER ENABLE=1; uma das saídas NAND vai para baixo, dependendo do estado de A.



Atrasos de Propagação

- Sempre que um sinal muda de estado na saída dos FFs, existe um atraso de tempo a partir do instante em que **o sinal é aplicado** até o instante em que **a saída comuta de estado**.
- Os **atrasos de propagação** são medidos em relação à borda de subida na entrada *CLK*.
- São medidos entre os pontos de 50 por cento da amplitude das formas de onda do clock e da saída.

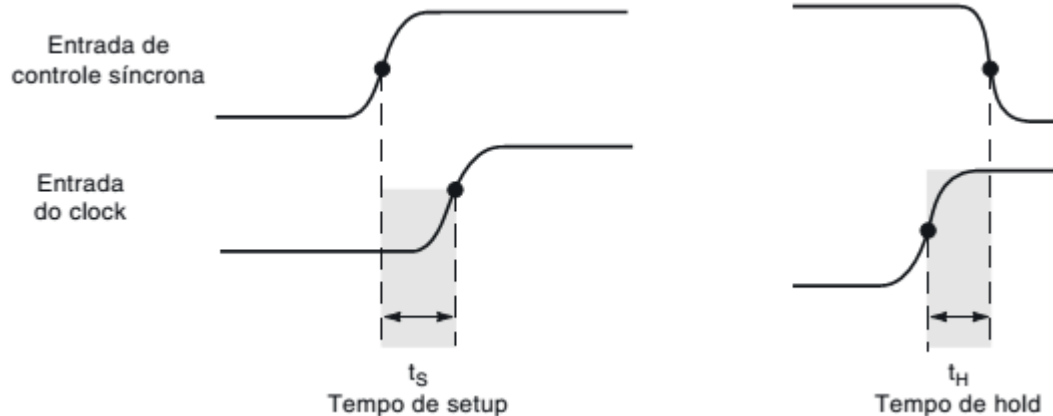
Atraso De Propagação nos Contadores Assíncronos

- Em contadores assíncronos, cada FF é disparado pela transição de saída do precedente.
- Cada FF adiciona um atraso de propagação (t_{pd}) na sua saída.
- Como a entrada do n-esimo FF depende da saída do anterior e supondo que os FF's do circuito tem atrasos constantes, o atraso do n-esimo vai depender do número de FF's do que antecederem ele:

$$t_{pdn} = (n-1) * t_{pd}$$

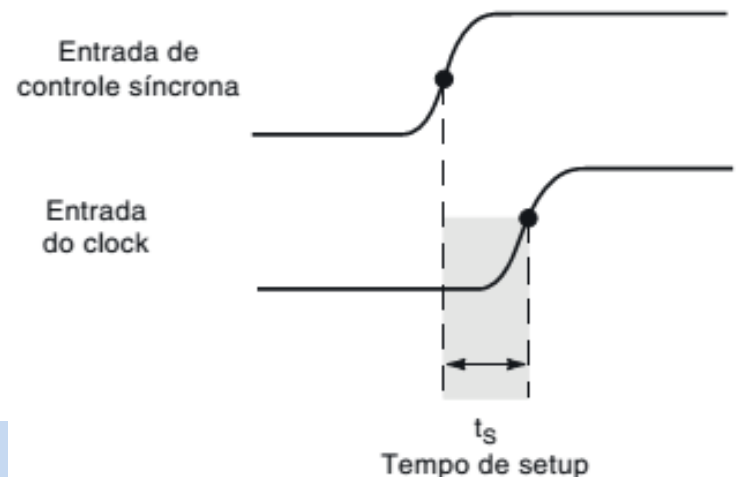
Parâmetros de Temporização

- Dois parâmetros devem ser observados para que o FF com clock responda de forma confiável às entradas de controle quando ocorrer uma transição ativa da entrada CLK
 - Tempo de Setup (t_s) (preparação)
 - Tempo de Hold (t_h) (manutenção)



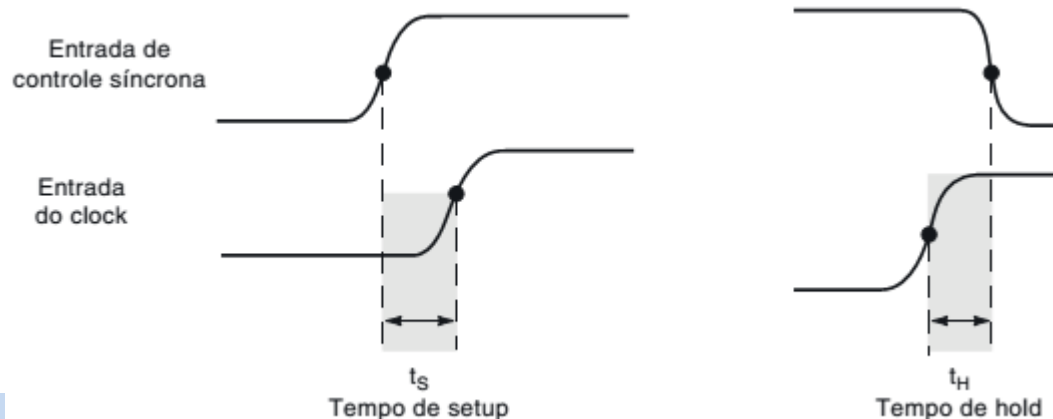
Parâmetros de Temporização

- Tempo de Setup (t_s) (preparação)
 - Intervalo de tempo **que precede** imediatamente a transição **ativa** do sinal de clock durante o qual a entrada de controle deve ser mantida no nível adequado.
 - O tempo é medido entre os instantes em que as transições estão em 50%



Parâmetros de Temporização

- Tempo de Hold (t_h) (manutenção)
 - Intervalo de tempo **que segue** imediatamente após a transição ativa do sinal de clock durante o qual a entrada de controle deve ser mantida.
 - Fabricantes determinam este valor e se não respeitado o FF pode responder de forma não confiável.

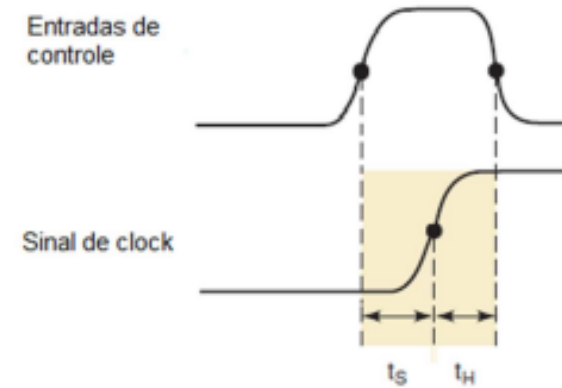
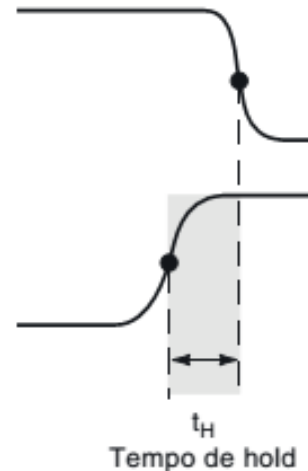
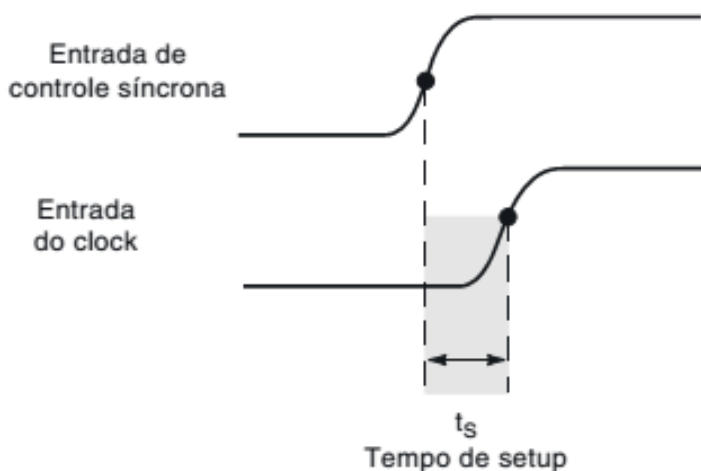


Parâmetros de Temporização

- Para garantir que o FF funcione corretamente quando ocorrer uma transição ativa do clock
 - Entradas de controle não devem mudar de estado por pelo menos 1 intervalo de tempo $t_{s(min)}$ antes da transição de clock.
 - Entradas de controle não devem mudar de estado por pelo menos 1 intervalo de tempo $t_{h(min)}$ após a transição de clock.
- Estes intervalos são necessários devido aos atrasos de propagação das portas internas que controlam a operação dos dispositivos de flip-flop.

Parâmetros de Temporização

- Valores típicos:
 - Tempo de Setup (t_s) (preparação)
 - Valores mínimos na ordem de 5 a 50ns
 - Tempo de Hold (t_h) (manutenção)
 - Valores mínimos na ordem de 0 a 10ns
 - Tempos são medidos entre os instantes em que as transições estão em 50%



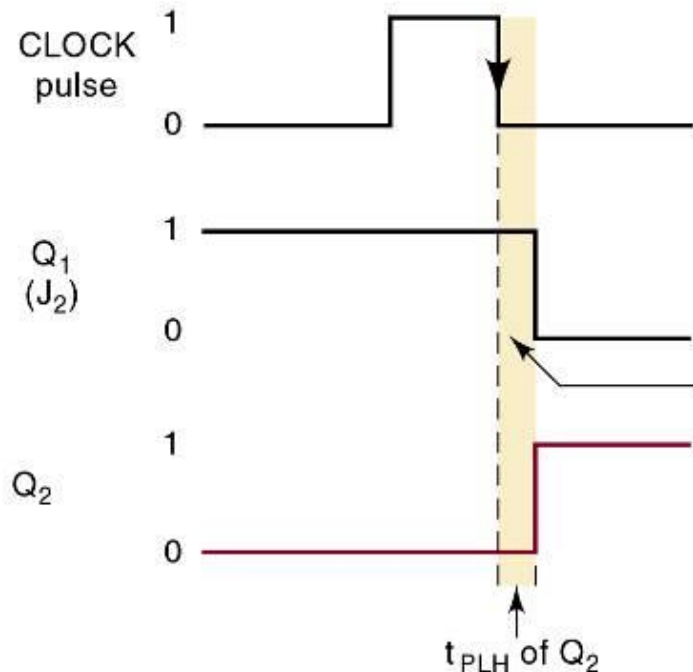
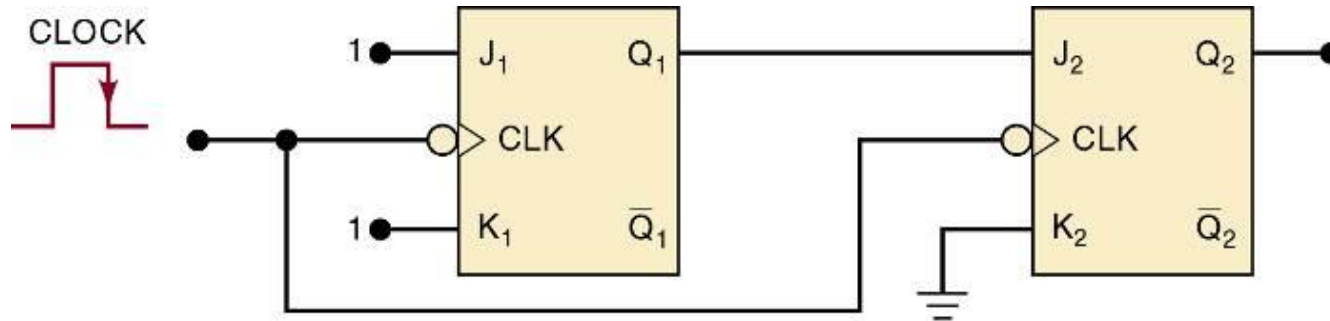
Problemas Temporais Potenciais em Circuitos com Flip-Flops

- Quando a saída de um FF está conectada a entrada de outro FF e ambos são disparados pelo mesmo clock, existe um potencial problema de temporização
 - O atraso de propagação pode provocar saídas imprevisíveis
- FF disparados por borda têm requisitos de tempo de manutenção (*hold time*) de 5 ns ou menos → maioria $t_H=0$

Assuma que o hold time do FF é curto o suficiente para responder confiavelmente de acordo com a seguinte regra:

A saída do Flip-Flop vai para um estado determinado pelos níveis lógicos presentes nas entradas de controle síncronas exatamente antes à transição ativa do clock.

Problemas Temporais Potenciais em Circuitos com Flip-Flops

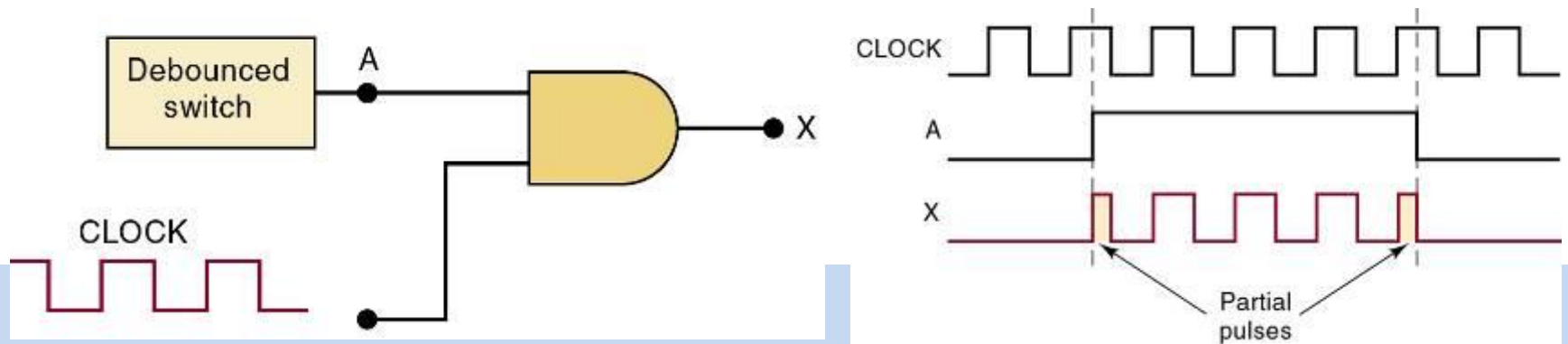


Q2 vai responder adequadamente ao nível presente em Q1 antes da transição negativa (NGT) do CLK \rightarrow desde que o hold time de Q2, t_H , seja menor que o atraso de propagação de Q1

Sincronização de Flip-Flops

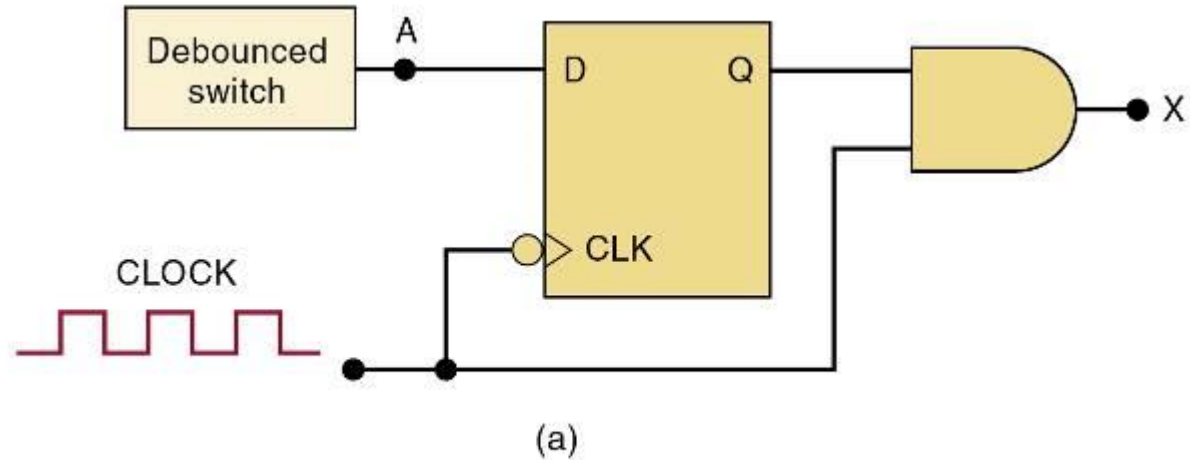
- Maioria dos sistemas opera de maneira síncrona — a mudança do estado depende do *clock*
- Operação assíncrona e síncrona são comumente combinadas
 - frequentemente através de entradas dadas por um humano
 - A natureza aleatória das entradas assíncronas *pode* resultar em saídas inesperadas

O sinal assíncrono em A pode resultar em pulsos parciais em X.



Sincronização de Flip-Flop

O Flip-Flop D disparado por
borda sincroniza a ativação
da porta **AND** à transição
negativa do *clock*



Vimos essa aplicação
quando falamos de F.F. D !

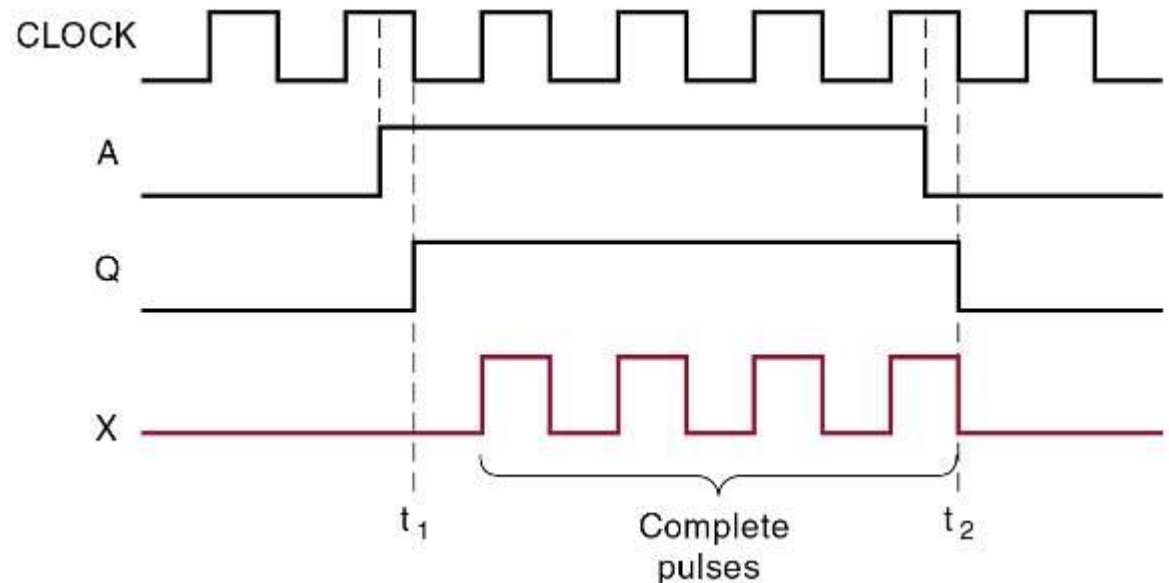
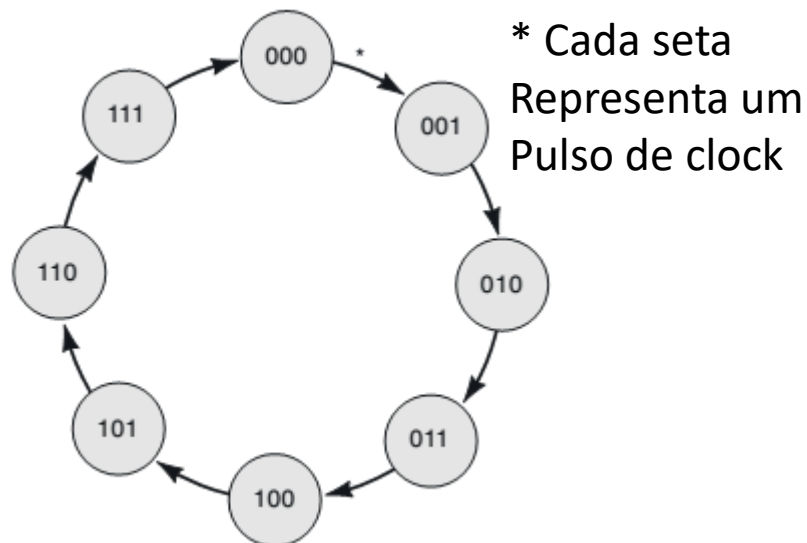


Diagrama de Transição de Estados

- Mostra o estado dos FFs a cada pulso de clock.
- O círculo contendo o número binário 100 representa o estado 4 (ou seja, $Q_2 = 1$, $Q_1 = Q_0 = 0$).
Pode ser verificado o estado anterior e posterior dos FFs

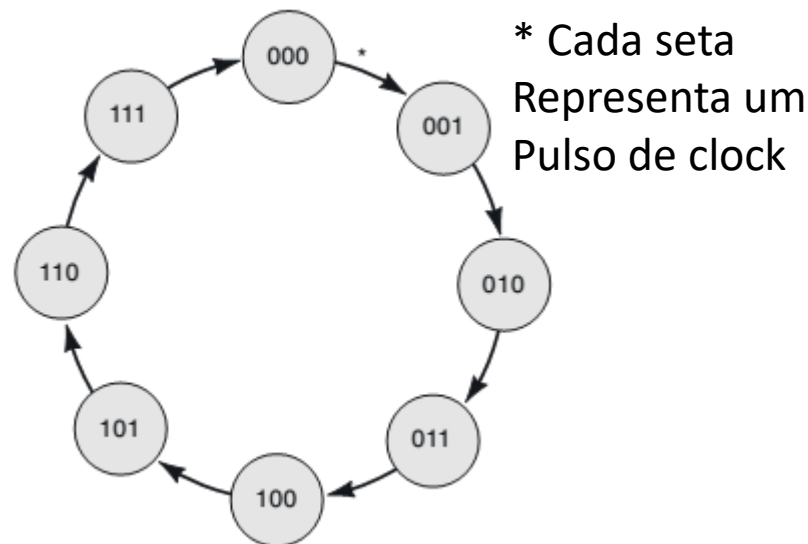


Módulo do Contador

- O módulo indica o número de estados da sequência de contagem.
- O modulo é definido pelo número de FF's.
- Se N flip-flops estão conectados na configuração anterior, o contador resultante terá 2^N estados diferentes e, portanto, será um contador de **módulo 2^N** .
- O contador anterior tem $2^3=8$ estados diferentes.
- O valor do módulo de um contador também indica a razão entre a frequência de entrada e a obtida na saída do último flip-flop ($1/2^N$) .

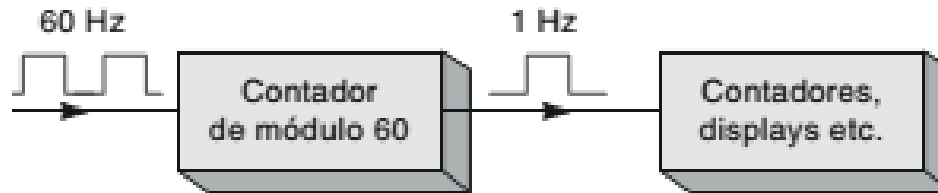
Módulo do Contador

- Considere que o contador de módulo 8 mostrado na Figura esteja no estado 101. Qual será o estado (a contagem) após a aplicação de 13 pulsos?



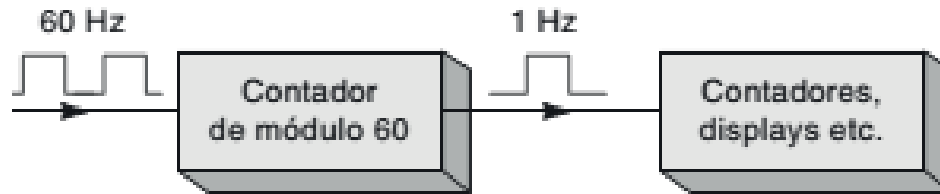
Exercício

- Uma onda quadrada de 60 Hz é colocada na entrada de um contador de módulo 60, usado para dividir a frequência de 60 Hz exatamente por 60, gerando uma forma de onda de 1 Hz. Qual a frequência de saída do circuito?



Exercício

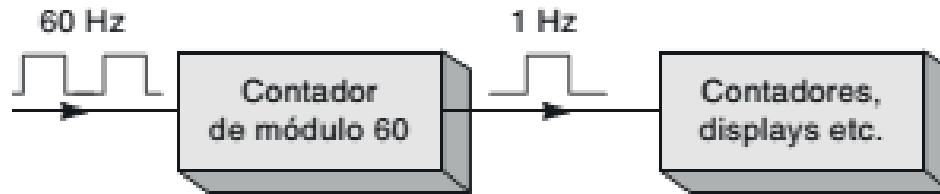
- Uma onda quadrada de 60 Hz é então colocada na entrada de um contador de módulo 60, usado para dividir a frequência de 60 Hz exatamente por 60, gerando uma forma de onda de 1 Hz.



$$freq_{final} = freq_{inicial} (1/2)^N \Rightarrow 1 = 60/2^N \Rightarrow N = 6 \Rightarrow 2^N = 64 \text{ Hz}$$

Exercício

- Uma onda quadrada de 60 Hz é então colocada na entrada de um contador de módulo 60, usado para dividir a frequência de 60 Hz exatamente por 60, gerando uma forma de onda de 1 Hz.



$$freq_{final} = freq_{inicial} (1/2)^N \Rightarrow 1 = 60/2^N \Rightarrow N = 6 \Rightarrow 2^N = 64 \text{ Hz}$$

- O contador deve ser resetado quando alcança a contagem 60 (111100)

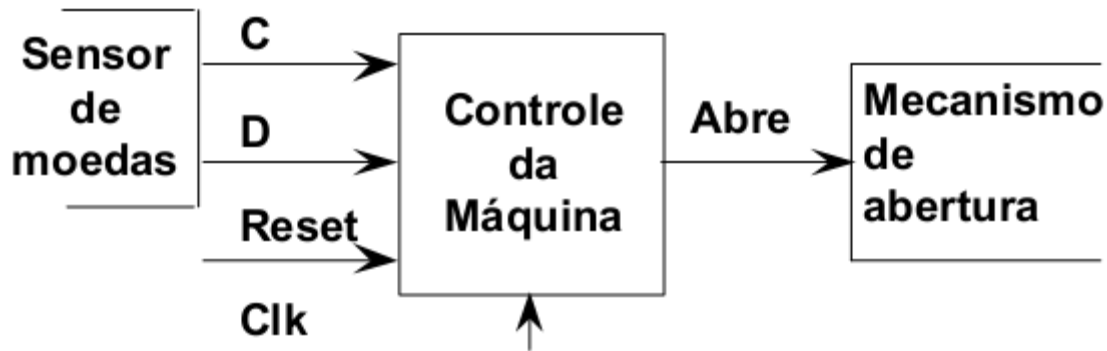
Projeto de Circuitos Sequenciais

- Algoritmo Básico
 - Entendimento do Problema
 - Análise das Informações
 - Diagrama de Estados
 - Minimização de Estados
 - Identificação de Estados
 - Escolha dos Elementos de Memória
 - Implementação da Lógica Combinacional
 - Minimização da Função
 - Redução do Circuito

Exemplo

Máquina de Venda

- **Funcionamento Geral**
 - Entrega um chiclete para cada 15 centavos
 - Entrada de moedas de 10 (D) e cinco (C) centavos
 - Não dá troco
- **Passo 1: Entendimento do problema**



Exemplo

Máquina de Venda de Chiclete

- Passo 2:
Análise das
Informações

Possibilidades :

C + C + C
C + D
D + C
D + D
C + C + D

Exemplo

Máquina de Venda de Chiclete

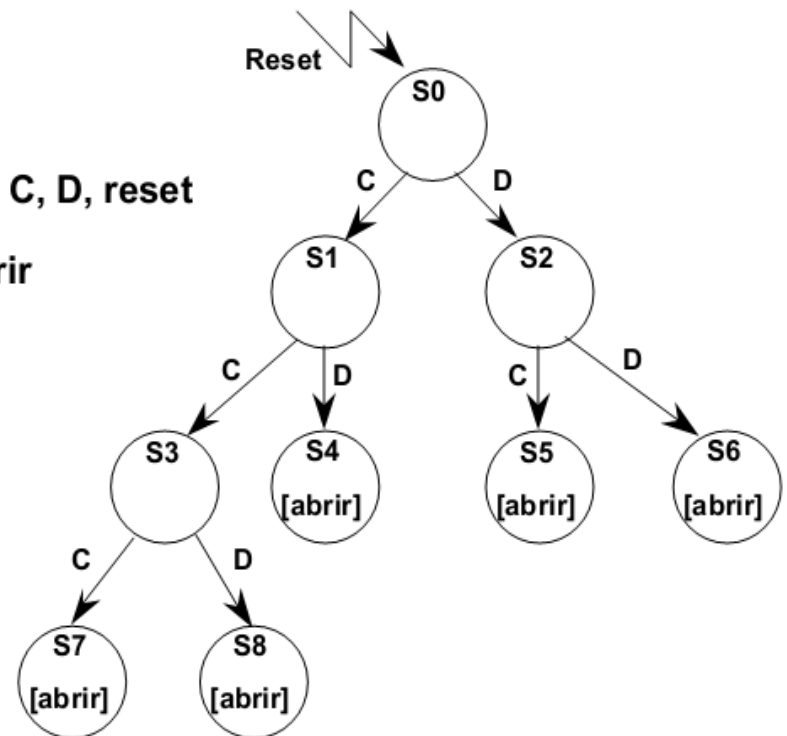
- Passo 2:
Análise das
Informações

Possibilidades :

C + C + C
C + D
D + C
D + D
C + C + D

Entradas: C, D, reset

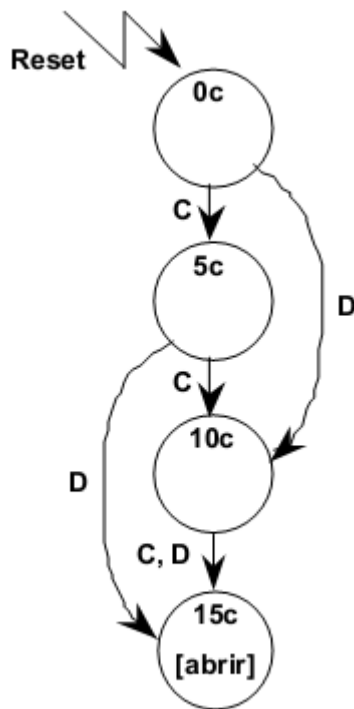
Saída: abrir



Exemplo

Máquina de Venda de Chiclete

- Passo 3.1: Minimização de estados
 - Reutilização de estados



Estado Atual	Entradas		Próximo Estado	Saída abrir
	D	C		
0c	0	0	0c	0
	0	1	5c	0
	1	0	10c	0
	1	1	X	X
5c	0	0	5c	0
	0	1	10c	0
	1	0	15c	0
	1	1	X	X
10c	0	0	10c	0
	0	1	15c	0
	1	0	15c	0
	1	1	X	X
15c	X	X	15c	1

Exemplo

Máquina de Venda de Chiclete

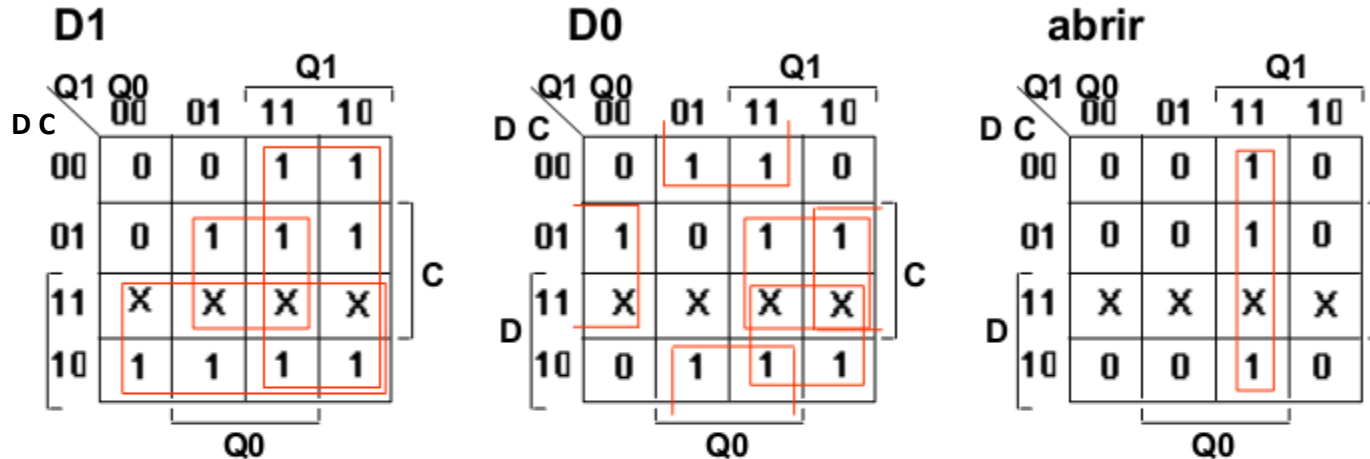
- Passo 3.2 e 3.3: Identificação de Estados e Escolha de Flip-Flops

Estado atual Q ₁ Q ₀		Entradas D C		Prox. estado D ₁ D ₀		Saída abrir
0	0	0	0	0	0	0
		0	1	0	1	0
		1	0	1	0	0
		1	1	X	X	X
0	1	0	0	0	1	0
		0	1	1	0	0
		1	0	1	1	0
		1	1	X	X	X
1	0	0	0	1	0	0
		0	1	1	1	0
		1	0	1	1	0
		1	1	X	X	X
1	1	0	0	1	1	1
		0	1	1	1	1
		1	0	1	1	1
		1	1	X	X	X

Exemplo

Máquina de Venda de Chiclete

- Passo 4.1: Minimização das Funções Lógicas



$$D1 = Q1 + D + Q0 C$$

$$D0 = C'Q0 + Q0'C + Q1 C + Q1 D$$

$$\text{abrir} = Q1 Q0$$

Estado atual Q ₁ Q ₀	Entradas D C	Prox. estado D ₁ D ₀	Saída abrir
0 0	0 0	0 0	0
	0 1	0 1	0
	1 0	1 0	0
	1 1	X X	X
0 1	0 0	0 1	0
	0 1	1 0	0
	1 0	1 1	0
	1 1	X X	X
1 0	0 0	1 0	0
	0 1	1 1	0
	1 0	1 1	0
	1 1	X X	X
1 1	0 0	1 1	1
	0 1	1 1	1
	1 0	1 1	1
	1 1	X X	X

Exercícios

- Considere um circuito de um contador que possui seis FFs conectados, (isto é, $Q_5, Q_4, Q_3, Q_2, Q_1, Q_0$).
 - a) Determine o módulo do contador. R: 64
 - b) Determine a frequência na saída do último FF (Q_5) quando a frequência do clock de entrada for de 1 MHz. R: 15,625 kHz
 - c) Qual é a faixa de estados de contagem desse contador? R: 000000_2 a 111111_2 (0 a 63_{10})
 - d) Considere como estado (contagem) inicial o valor 000000. Qual será o estado do contador após 129 pulsos? R: 000001